Philipps-Universität Marburg

Image Segmentation of Bacterial Cells in Biofilms

Dissertation

zur Erlangung des Doktorgrades
der Naturwissenschaften
(Dr. rer. nat)

dem Fachbereich Physik
der Philipps-Universität Marburg
vorgelegt von

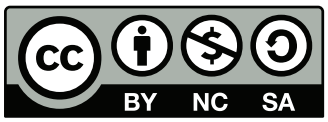
Eric Jelli

geboren in
Gießen

Marburg/Lahn, 2020

Vom Fachbereich Physik der Philipps-Universität als Dissertation angenommen am: 10.02.2021
Erstgutachter: Prof. Dr. Knut Drescher
Zweitgutachter: Prof. Dr. Victor Sourjik
Prüfer: Prof. Dr. Peter Lenz

Tag der mündlichen Prüfung: 19.02.2021
Hochschulkennziffer: 1180



Zusammenfassung

Bakterielle Biofilme sind drei-dimensionale Zellcluster, welche ihre eigene Matrix produzieren. Die selbst-produzierte Matrix bietet den Zellen einen gemeinschaftlichen Schutz vor äußeren Stressfaktoren. Diese Stressfaktoren können abiotischer Natur sein wie z.B. Temperatur- und Nährstoffschwankungen, oder aber auch biotische Faktoren wie z.B. Antibiotikabehandlung oder Bakteriophageninfektionen. Dies führt dazu, dass einzelne Zelle innerhalb der mikrobiologischen Gemeinschaften eine erhöhte Widerstandsfähigkeit aufweisen und eine große Herausforderung für Medizin und technische Anwendungen darstellen. Um Biofilme wirksam zu bekämpfen, muss man die dem Wachstum und Entwicklung zugrundeliegenden Mechanismen entschlüsseln.
Aufgrund der hohen Zelldichte innerhalb der Gemeinschaften sind die Mechanismen nicht räumlich und zeitlich invariant, sondern hängen z.B. von Metabolit-, Nährstoff- und Sauerstoffgradienten ab. Daher ist es für die Beschreibung unabdingbar Beobachtungen auf Einzelzellebene durchzuführen. Für die nicht-invasive Untersuchung von einzelnen Zellen innerhalb eines Biofilms ist man auf konfokale Fluoreszenzmikroskopie angewiesen. Um aus den gesammelten, drei-dimensionalen Bilddaten Zelleigenschaften zu extrahieren, ist die Erkennung von den jeweiligen Zellen erforderlich. Besonders die digitale Rekonstruktion der Zellmorphologie spielt dabei eine große Rolle. Diese erhält man über die Segmentierung der Bilddaten. Dabei werden einzelne Bildelemente den abgebildeten Objekten zugeordnet. Damit lassen sich die einzelnen Objekte voneinander unterscheiden und deren Eigenschaften extrahieren.
Im ersten Teil dieser Arbeit wird ein benutzerfreundliches Computerprogramm vorgestellt, welches die Segmentierung und Analyse von Fluoreszenzmikroskopiedaten wesentlich vereinfacht. Es stellt eine umfangreiche Auswahl an traditionellen Segmentieralgorithmen, Parameterberechnungen und Visualisierungsmöglichkeiten zur Verfügung. Alle Funktionen sind ohne Programmierkenntnisse zugänglich, sodass sie einer großen Gruppe von Benutzern zur Verfügung stehen. Die implementierten Funktionen ermöglichen es die Zeit zwischen durchgeführtem Experiment und vollendeter Datenanalyse signifikant zu verkürzen. Durch eine schnelle Abfolge von stetig angepassten Experimenten können in kurzer Zeit schnell wissenschaftliche Einblicke in Biofilme gewonnen werden.
Als Ergänzung zu den bestehenden Verfahren zur Einzelzellsegmentierung in Biofilmen, wird eine Verbesserung vorgestellt, welche die Genauigkeit von bisherigen Filter-basierten Algorithmen übertrifft und einen weiteren Schritt in Richtung von zeitlich und räumlich aufgelöster Einzelzellverfolgung innerhalb bakteriellen Biofilme darstellt.
Abschließend wird die Möglichkeit der Anwendung von Deep Learning Algorithmen für die Segmentierung in Biofilmen evaluiert. Dazu wird eine Methode vorgestellt welche den Annotationsaufwand von Trainingsdaten im Vergleich zu einer vollständig manuellen Annotation drastisch verkürzt. Die erstellten Daten werden für das Training von Algorithmen eingesetzt und die Genauigkeit der Segmentierung an experimentellen Daten untersucht.

Abstract

Bacterial biofilms are three-dimensional cell communities that live embedded in a self-produced extracellular matrix. Due to the protective properties of the dense coexistence of microorganisms, single bacteria inside the communities are hard to eradicate by antibacterial agents and bacteriophages. This increased resilience gives rise to severe problems in medical and technological settings. To fight the bacterial cells, an in-detail understanding of the underlying mechanisms of biofilm formation and development is required. Due to spatio-temporal variances in environmental conditions inside a single biofilm, the mechanisms can only be investigated by probing single-cells at different locations over time. Currently, the mechanistic information is primarily encoded in volumetric image data gathered with confocal fluorescence microscopy. To quantify features of the single-cell behaviour, single objects need to be detected. This identification of objects inside biofilm image data is called segmentation and is a key step for the understanding of the biological processes inside biofilms.
In the first part of this work, a user-friendly computer program is presented which simplifies the analysis of bacterial biofilms. It provides a comprehensive set of tools to segment, analyse, and visualize fluorescent microscopy data without writing a single line of analysis code. This allows for faster feedback loops between experiment and analysis, and allows fast insights into the gathered data.
The single-cell segmentation accuracy of a recent segmentation algorithm is discussed in detail. In this discussion, points for improvements are identified and a new optimized segmentation approach presented. The improved algorithm achieves superior segmentation accuracy on bacterial biofilms when compared to the current state-of-the-art algorithms.
Finally, the possibility of deep learning-based end-to-end segmentation of biofilm data is investigated. A method for the quick generation of training data is presented and the results of two single-cell segmentation approaches for eukaryotic cells are adapted for the segmentation of bacterial biofilm segmentation.

Table of Contents

Acronyms

2D two-dimensional

3D three-dimensional

CLSM confocal laser scanning microscope

CSV comma separated values

cyclic di-GMP cyclic diguanylate

DNA deoxyribonucleic acid

EM electron microscopy

EPS extracellular polymeric substance

FCS flow cytometry standard

FN false negative

FP false positive

GMM Gaussian mixture model

GPU graphic processing unit

GUI graphical user interface

HTML hypertext markup language

ID identifier

IoU intersection over union

LB lysogeny broth

LoG Laplacian of Gaussian

MCT maximum correlation thresholding

MEM minimum essential medium

MLE maximum likelihood estimation

MSDE mean square displacement error

NA numerical aperture

NaN not a number

NMS non-maximum suppression

OD optical density

PBS phosphate-buffered saline

PCA principal component analysis

PDMS Polydimethylsiloxan

PSF point spread function

RAM random access memory

ROI region of interest

SDCM spinning disk confocal microscope

sfGFP superfolder green fluorescent protein

SGD stochastic gradient descent

SNR signal-to-noise ratio

TIFF tagged image file format

TP true positive

VTK visualization toolkit

1 Introduction

For a long time, bacterial cells were believed to strictly constitute an individual lifestyle [34]. Based on recent estimates, however, a predominant form of bacterial life is driven by close interaction of individual cells in proximity [50]. First hints of the collaborative nature of bacterial cells were revealed by the discovery of autoinducers which foster the inter-cellular communication [121]. Single bacterial cells are able to modify their behaviour based on the kin abundance in proximity [52].

The insight that multicellular communities dominate the bacterial life on Earth, resulted in a growing interest in the investigation of the complex interaction among single cells in dense, three-dimensional (3D), multi-cellular communities, termed biofilms [50]. The gene expression of bacteria inside biofilms is not only different from their planktonic counterparts, but also exhibits heterogeneity within the same micro colony due to different nutrient availability and waste product accumulation [141, 155]. This results in bacterial strongholds, which have an enhanced tolerance against antibiotics and lead to chronic infections that are hard to defeat [110]. Besides the hazard of biofilm development for human health, biofilms also undermine the function of machinery and grow in fresh water systems [47, 133]. Understanding the underlying principles of biofilm development is important to develop new treatment approaches against biofilm infections and has the potential to increase the efficiency of water bearing components in industry.

The influence of the spatial gradients of nutrients, waste products, and signalling molecules on the behaviour of single cells remains largely unclear [40]. To discover new approaches against the increased antibiotic tolerance [125], a deep understanding of the underlying mechanisms is required. To understand the regulatory patterns, the spatial arrangement of different genotypes within the microbial community needs to be uncovered [120]. For the explanation of biofilm-specific properties, it is necessary to understand the physiological changes in a single cell with respect to its spatial position inside and the environment outside of the biofilm. For this comparison single-cell resolution is needed [183]. Currently, only digital image cytometry is capable to analyse these patterns in vitro [64]. Capturing biofilms at single-cell resolution and the analysis of the gathered data, however, remains challenging.

1.1 Aims of this Thesis

The introduction of automatic microscopes resulted in an abundance of image data. Today, imaging facilities acquire large datasets, which need to be converted into storable file formats, reviewed, processed, and analysed [21]. The amount of collected data is further increased by modern confocal laser scanning microscopes (CLSMs), spinning disk confocal microscopes (SDCMs), and light sheet microscopes which can capture 3D volumes instead of two-dimensional (2D) images with similar or even higher temporal resolution.

In the past, the image acquisition equipment was the bottleneck in the experiment pipeline. Today, image analysis becomes more and more the limiting factor. While the options for sophisticated biofilm experiments increased, the analysis software did not keep track. The still widely used *Comstat* [72] program for the analysis of 3D volumes lacks flexibility and options to describe important biofilm properties with 3D spatial resolution inside dense communities. At the same time highly versatile programs for the analysis of microscope data like *Fiji* [142] have – due to a large community of open source developers – steadily improved. Yet, the analysis of images is bound to *Fiji*-specific macro language which is not commonly used in the field of image analysis outside biology. Thus, the effective usage of *Fiji* for large batch processing still requires at least basic programming skills in a highly specific language.

Given the current needs of the biofilm research community, a versatile program for the analysis of biofilm microscope data is needed. Thus, the development of a graphical user interface (GUI) program was started. The resulting software was termed „*BiofilmQ*“ and provides an all-in-one solution for image analysis, quantitative measurements, and graphical presentation required in a typical image analysis workflow (Sec. 3).

At the same time, limitations in the current state-of-the-art segmentation algorithms became apparent. The current single-cell segmentation approaches are not suitable for the tracking of single cells in a developing and growing biofilm. The achievable segmentation accuracy is not high enough to unambiguously identify single cells in consecutive time frames. While recent publications employed fluorescent foci to capture patterns of moving bacteria inside biofilms [134], it is unknown how lineage trees for every single cell inside a biofilm can be captured. To reduce the amount of genetic modifications to the organism, a lineage reconstruction based on constitutive fluorophore production would be a preferred solution. This requires highly accurate segmentation algorithms.

This thesis is structured as follows: In the upcoming Sec. 2, I introduce essential theoretical aspects for the segmentation of bacterial biofilms. In Sec. 3, I introduce the analysis software which was created as part of my work in the research group of Prof. Dr. Knut Drescher. This section is extended with a short section (Sec. 4) about an improved segmentation algorithm in the *BiofilmQ* framework which results in superior segmentation results in dense biofilms. As shown in Sec. 5, the extension achieves superior segmentation accuracy when compared with the current state-of-the-art classical segmentation pipeline, but is out-performed when compared with deep learning segmentation approaches. Thus, Sec. 5 contains a description of an iterative data annotation pipeline which can be used to create the required training data for current machine learning algorithms.

2 Image Analysis in Bacterial Biofilms

2.1 Bacterial Biofilms

The imaging (and the sub-sequential image analysis) of bacterial biofilms with a confocal microscope is a challenging task. In particular, three physical obstacles make the image acquisition difficult: (1) Length scale, (2) light sensitivity, and (3) optical properties of the specimen.

Length scale Typical model organisms for biofilm formation are *Escherichia coli*, *Pseudomonas aeruginosa*, and *Vibrio cholerae*. The diameter of those bacteria range from $\sim 250\,\text{nm}$ to $\sim 500\,\text{nm}$. These length scales are close to the lateral resolution and even lower than the axial resolution of a confocal microscope (Sec. 2.2 and [183]). Deconvolution (Sec. 2.5) can lower the diffraction limited length scale and is commonly used to achieve single-cell resolution in biofilm experiments [179, 134].

Light sensitivity Current confocal microscopes illuminate not a single point inside the specimen, but a light cone which stretches below and above the focal point [132]. This results in a large photon exposure of the imaged specimen; sometimes close to the limit of photobleaching and phototoxicity. Recent studies, which investigated spatial properties inside bacterial biofilms at single-cell resolution [151, 66, 38], are very close to the biological limit. Only with sophisticated adaptive microscope control software, the light exposure of the specimen can be reduced below the physiological limit. The remaining signal intensity requires electron multiplied charge-coupled device cameras which can capture very low light emissions. Only with this equipment it is possible to capture time-lapse images for the full biofilm development cycle on a spinning disk confocal microscope (SDCM) with the required temporal and spatial resolution [66]. Advanced post-processing techniques such as content aware restoration [171] can be used to reduce the light exposure and enable imaging with high temporal resolution. However, no application of the technique on biofilms has been demonstrated yet. Light sheet microscopy is another image acquisition approach which can reduce photobleaching and -toxicity. Instead of illuminating the sample with a light cone, it uses a selective plane and captures the light emission from the whole plane at once [79, 134].

The intensity of the excitation beam cannot be arbitrarily reduced. The physical limitations arise due to the reduced penetration depth of the excitation beam and the light attenuation by the biofilm matrix in the specimen [13].

Optical properties The refractive indices inside bacterial biofilms differ spatially within the specimen and are different from the embedding medium [132]. Currently, biofilm experiments with single-cell resolution are conducted in microfluidic devices [41]. The biofilms are completely submerged in the water-based

medium. Therefore, water or silicone oil objectives with matching refractive index of the embedding medium and the objective immersion medium are used. This prevents spherical aberrations and benefits the resulting image quality [134, 66]. Even with specialized objectives, the problem of optical heterogeneity inside the biofilm remains unsolved.

With sophisticated confocal microscopes and suitable post-processing algorithms it becomes possible to image (and segment) biofilms with single-cell resolution. In the following subsections, I would like to introduce some theoretical background on the optical imaging systems, the building blocks of image processing, and the different approaches for the segmentation of single cells inside biofilms.

2.2 Confocal Microscopy

Besides the availability of magnetic resonance imaging or scanning transmission X-ray microscopy, the principal approach for the investigation of dense, volumetric microbial communities remains the optical microscopy [124]. The diameter of biofilm-forming bacteria are just above the physical resolution limits of traditional wide-field microscopes. According to Abbe [1], the optical resolution d_{min} of a microscope is diffraction-limited according to

$$d_{\text{min}} = \frac{\lambda}{2\,\text{NA}}, \tag{1}$$

where λ is the wavelength of the captured signal and the numerical aperture (NA) is given by

$$NA = n \cdot \sin(\theta). \tag{2}$$

The NA depends on the half-angle of the focused light cone of the objective θ and the refraction index of the objective immersion medium n (i.e. $n_{\text{oil}} = 1.518$, $n_{\text{silicon oil}} = 1.406$, $n_{\text{water}} = 1.33$, $n_{\text{air}} \sim 1.0$).

Based on (2), the resolution of an optical microscope in lateral direction (or in the focus plane, in the following denoted with the x- and y-coordinates) d_{xy} is described by the Rayleigh-criterion [108, 132]

$$d_{xy} = \frac{1.22\,\lambda}{2\,\text{NA}}, \tag{3}$$

which describes the minimum optical-resolvable spacing. The optical resolution of a traditional wide-field microscope in axial direction (i.e. orthogonal to the focus plane) d_{axial} is given by

$$d_{\text{axial}} = \frac{2\lambda\eta}{(\text{NA})^2}, \tag{4}$$

where η denotes the refractive index of the embedding medium of the specimen [132].

Confocal microscopes have an improved axial resolution in comparison to wide-field microscopes. In a confocal microscope out-of-focus light is blocked by one

or more small pinholes. The pinholes increase the optical sectioning and enable the quantification of fluorescent structures with high spatial precision [85]. Light, which does not emerge from the illumination focal point, is rejected. Today, two major implementation of confocal microscopes exist: Confocal laser scanning microscopes CLSMs and SDCMs.

CLSMs use galvo-mounted mirrors to scan the specimen one spot at a time. A variable pinhole size enables high lateral resolution. Additional features include area-specific illumination and a variable distance between the scanning points. The light emitted by the specimen is usually captured by a point detector. Due to the pixel-by-pixel scanning approach and a pixel dwelling time of $\sim 1\,\mu s$, imaging a single plane can take ~ 1 - $2\,s$. Small dwelling times need high light intensity which increases the light-induced stress on the specimen. Scanning large volumes – such as biofilms – limit the temporal resolution even further.

In contrast, SDCMs use a spinning disk with an array of pinholes to achieve confocality (Fig. 1). Typical pinhole sizes are between $25\,\mu m$ and $75\,\mu m$. In a typical set up, 1000 pinholes are illuminated at once and the signal is directly captured by a camera. This technique comes with the advantages of fast confocal imaging, slower bleaching rate, and less phototoxicity when compared to point scanning systems [132]. The multiple pinholes, however, can enable pinhole-cross talk. Pinhole-cross talk describes the effect when scattered light from other points than focal plane reaches the camera sensor. It increases the background signal for thicker specimen such as biofilms and thus negatively affects the signal-to-noise ratio (SNR).

While the lateral resolution of a SDCM is the same as that of a traditional wide-field microscope (3), the axial resolution d_z is improved [132] according to

$$d_z = \frac{0.67 \cdot \lambda}{\left(n - \sqrt{n^2 - \mathrm{NA}^2}\right)}. \tag{5}$$

This enables the spatial resolution of individual fluorescent cells inside bacterial biofilms.

To achieve the highest spatial resolution possible, high NA objectives have to be used. Due to the higher refraction index, oil immersion objectives achieve very high NA. Yet, they only allow for short working distances (usually $< 200\,\mu m$). With a typical cover slip thickness of $170\,\mu m$ they can only image small biofilms.

Recent studies [134] used dual-view light-sheet microscopy to overcome the limitations in the axial direction of SDCMs in biofilm imaging. This relatively new light sheet microscope technique has been reported to achieve isotropic resolution in all three spatial directions [178, 97].

Modern optical microscopes can even resolve structures which are smaller than the diffraction limit. These microscopes use super-resolution techniques such as structured illumination microscopy [70] , stimulated emission depletion (STED) [71], single molecule localization microscopy [102], or MINFLUX [11]. Super-

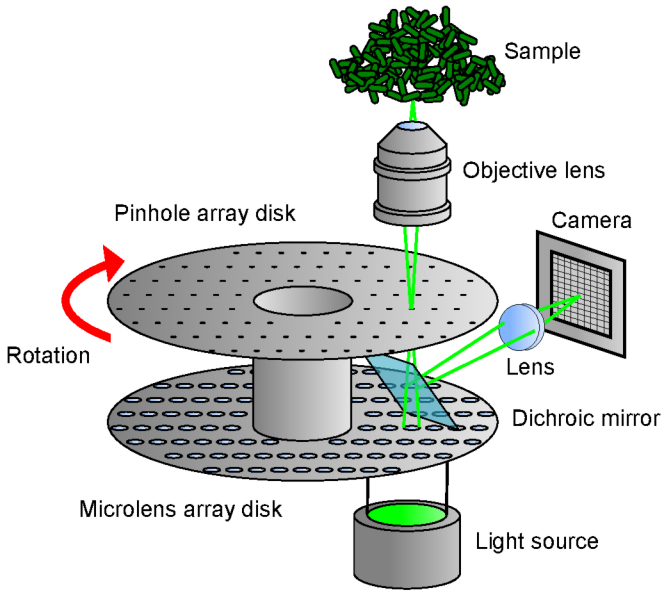


Figure 1: Schematic overview of a spinning disk confocal microscope (SDCM). A light source emits a collimated excitation laser beam. An array of rotating micro lenses focuses the excitation beam on a co-rotating array of pinholes. Light, which passes through micro lenses and pinholes, is focused on the sample by the objective. In the sample, the incoming light excites fluorophores which themselves emit light with a higher wavelength. Due to the wavelength shift, the emitted light cannot pass the dichroic mirror; is thus deflected, and passes through an emission filter and a lens. The lens focuses the light on the camera sensor. Light which passes different pinholes in the pinhole array disk is focused in different locations on the camera sensor resulting in the simultaneous detection of a large number of pinhole rays.

resolution microscopes, however, come with additional limitations concerning the sample preparation and image acquisition.

The given limits for the axial (3) and lateral (5) resolution are the optical limits of the used equipment. Especially for low intensity images, quantum effects can degrade the measurement accuracy further. The photo effect of single photons is such a quantum effect and follows a Poisson statistic [132]. That means, that uncertainty in the counted photon events at a pixel on the image sensor N is given by $\sqrt{N}$. Therefore, a high photon signal is desired, since it decreases the relative uncertainty in the measured pixel intensity. Photon shot noise is part of the intrinsic noise during an experiment and cannot be avoided by better alignment of the imaging system.

During the conversion from analogue (continuous) to digital (discrete) signal, sampling errors can occur. This can be avoided by choosing sampling distances (i.e. the distance between consecutive images in z-direction) according to the Nyquist criterion. According to the Nyquist criterion, an analogue signal is optimally sampled if the interval between the sampling steps is less than half the smallest size of the smallest structure in the signal. In the case of confocal microscopy, the smallest structures are those at the diffraction limit.

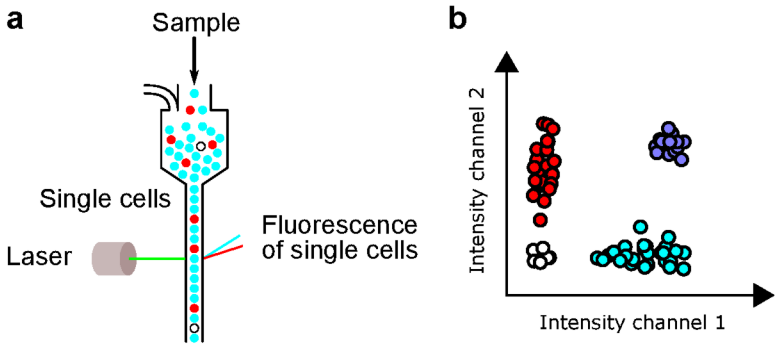


Figure 2: Schematic overview of a flow cytometer measurement. **a** Suspended single cells with fluorescent markers are prepared in phosphate-buffered saline (PBS) and flow through a thin focus region of the excitation lasers. The cells re-emit a signal upon excitation due to the markers. The emission can be quantitatively detected with an optical sensor. **b** Flow cytometer experiments are typically visualized in 2D scatter plots. Each point represents the emitted intensity of a single cell in the two indicated channels. In this schematic overview, four populations are visible. The red and turquoise populations represent the different cells types in the sample. Additionally, a population without marker signal, and a population with a combined signal is visible. Detections without marker signal arise due to cell debris (i.e. lysed cells which emerged during the sonication). The combined signal is created by doublets of cells with different markers which pass the laser focus together.

2.3 Flow Cytometry

Although closely related to fluorescence microscopy, flow cytometry does not preserve the spatial relation between the cells (Fig. 2). Rather, it requires the single cells to be suspended in a liquid medium [33, 3]. The cells are analysed by laser excitation. The forward and backward scattering of the laser intensity is collected. Additionally, the emission wavelengths of potentially excited fluorophores in the single-cells are captured. This allows the detection of the fluorescent signal of a single cell without the difficulties which arise due to varying refractive indices inside biofilms.

The flow cytometer analysis comes with the disadvantage of the sample preparation. Cells inside mature biofilms are attached to each other with extracellular polymeric substance (EPS) components [164]. Disrupting the matrix requires sonication (using sound to disrupt the bonds between the cells). Although commercial devices can analyse large numbers of cells within seconds, the data acquisition speeds are multiple orders of magnitude slower than the ones of a confocal microscope. Additionally, the loss of the spatial relation between the different expression levels renders it currently unusable for the description of spatial heterogeneous cell populations inside bacterial biofilms without sophisticated marker schemes.

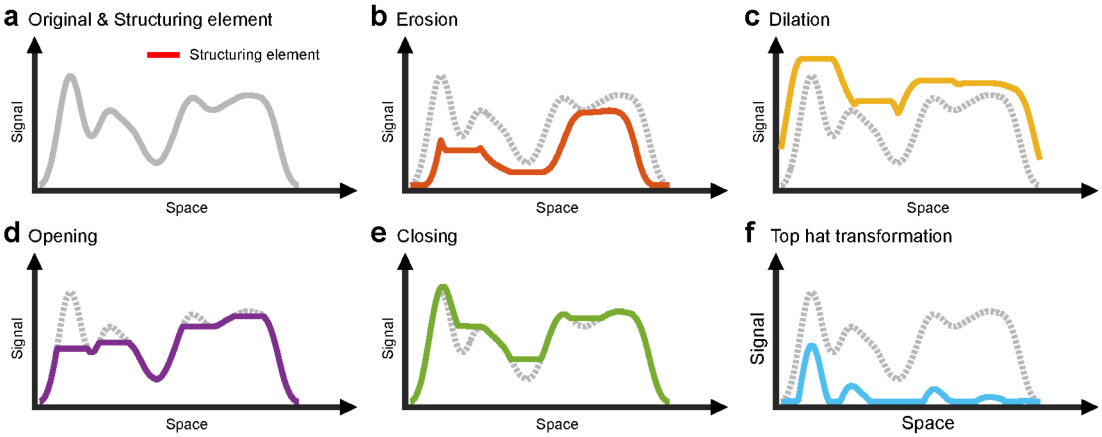


Figure 3: Example of morphological operations on a continuous signal in one dimension. **a** Original signal with variation in space and the structuring element. **b** Example of the erosion of the signal with the structuring element. **c** Resulting signal after a dilation. **d** The morphological opening of the signal is given by an erosion followed by a dilation. **e** A dilation and subsequent erosion of the signal results in the morphological closing. **f** The top hat transformation of the signal is given by the signal itself with the opening subtracted.

2.4 Image Processing

Microscopes are powerful tools to investigate microorganisms. Yet, the output images do not directly produce quantitative insights into the mechanisms of the observed systems. In the past, the qualitative description of changes of the system were sufficient to report new observations. Today, quantifiable features need to be extracted from the images to prove (or disprove) scientific hypotheses. To extract these features from the captured images, digital image processing is needed.

With the digitalization of the fluorescent signal another problem arises: Photon counting events are integrated over a small spatial area on the camera sensor and represented by single matrix entities called pixels. In the case of 3D volumes, the sampling size in z-direction adds an additional discretization step. The resulting discrete volume elements are called voxels [132, 119].

3D image volumes are created by combining 2D images into a stack. In the following, I introduce common image processing steps for the 2D domain. These image processing steps can be extended to the third dimension without loss of generality. The additional dimension, however, comes with increased requirements on the processing hardware in terms of compute resources and memory capacity.

One of the simplest image processing step is the application of a binary threshold. Here, all pixel intensities below or equal a threshold value are assigned to zero, all values above are assigned to one. Binary thresholding can be used as a very simple segmentation approach. More sophisticated segmentation methods will be explained in Sec. 2.6. Since manual thresholding easily results in analysis biases, automatic threshold approaches can increase the reproducibility of the image analysis [170].

One method to automatically determine an image threshold was introduced by Otsu in 1979 [128, 55]. The proposed algorithm maximizes the between-class variance of the threshold-divided classes. To calculate the between-class variance, a histogram of the image in L different intensity levels ($\{0, 1, 2, \ldots, L-1\}$) is created. Suppose the normalized histogram components are given by p_i, and the chosen threshold is denoted with k, then the probability $P_1(k)$ that a pixel value is smaller or equal k is given by

$$P_1(k) = \sum_{i=1}^{k} p_i, \tag{6}$$

and the cumulative mean of these pixel values can be calculated via

$$m(k) = \sum_{i=0}^{k} i p_i. \tag{7}$$

The sought threshold is defined as $k \in 0, 1, \ldots, L-1$ such that the between class variance σ_B^2, which is given by

$$\sigma_B^2(k) = \frac{[m(L-1) \cdot P_1(k) - m(k)]^2}{P_1(k)\,[1 - P_1(k)]}, \tag{8}$$

is maximized [55].

Another approach to modify image values are called image filters. An image filter describes an image processing step in which the input and the output are both images. Often, filters are used for image restoration (i.e. the removal of noise) or for image enhancement (i.e. preparing the image for a downstream image analysis algorithm to extract image features) [55].

Mean filters are one approach to reduce noise in an image. The arithmetic mean filter (in the following just mean filter) on an image f with the dimensions $x \times y$ by a domain S_{xy} with the dimensions $m \times n$ is given by

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}}^{m,n} f(s, t). \tag{9}$$

The mean filter results in a noise reduction in the output image $\hat{f}$ by blurring.

Another filter type which provide excellent noise-reduction with considerably less blurring than the mean filter (9) is the median filter

$$\hat{f}(x, y) = \underset{(s,t) \in S_{xy}}{\text{median}}\{f(s, t)\}. \tag{10}$$

Linear filtering operations – such as the mean filter – can be implemented via convolutions. A convolution is a mathematical operation denoted by $w(x, y) \circledast$

$f(x, y)$. Here, w denotes the convolution kernel with the dimensions $2a \times 2b$ which is applied to the input image f according to

$$w(x, y) \circledast f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t). \tag{11}$$

One popular kernel choice is the Gaussian kernel

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2}), \tag{12}$$

which blurs the image by accumulating information of the neighbourhood in the output pixel. The range of the neighbourhood is given by the variance σ.

Filters cannot only be used to reduce noise in images, but also can extract lower level features such as edges. The Marr-Hildreth edge detection approach [109] uses the Laplacian of Gaussian (LoG) as filter kernel. The LoG is defined as

$$\nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right] \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \tag{13}$$

The initial Gaussian filter reduces the intensity of structures at scales much smaller than the variance in the image. The Laplacian ∇^2 is – in contrast to the first order derivative $\nabla = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}$ – invariant to rotations. Thus, the operation responds equally to changes in any direction [55]. The Marr-Hildreth edge detection convolves the LoG kernel with the input image and identifies the zero crossings in the output. These zero-crossings represent edges in the original image.

Image processing operations are not limited to filters, but can also include morphological operations. In contrast to filters, morphological operations use additional input called structural elements to modify image data. Morphological operations are often used on binary images to enhance or suppress features of the signal. The operations can be adapted for the modification of grey-level images. Fig. 3 shows the application of morphological operations on a continuous one-dimensional signal [55]. Two basic morphological operations are called erosion (Fig. 3b) and dilation (Fig. 3c). An erosion shrinks the signal by the structuring element, while a dilation enhances the signal by the same morphological quantity.

The morphological opening of an image f (or the in Fig. 3d shown signal) by structuring element b is denoted by $f \circ b$. The morphological opening is the erosion of f by b (Fig. 3b), followed by a dilation with b again:

$$f \circ b = (f \ominus b) \oplus b \tag{14}$$

Similarly, the morphological closing of a signal

$$f \bullet b = (f \oplus b) \ominus b \tag{15}$$

is defined as dilation, followed by an erosion (Fig. 3e)

A very powerful morphological operation to reduce stray fluorescence in microscope images is the top hat transformation. It emerges when the opening of f is subtracted from f itself

$$T_{\text{hat}}(f) = f - (f \circ b), \tag{16}$$

and suppresses signal fluctuations on larger length scales than the structuring element. The top hat transformation does not modify signal fluctuations below the kernel size.

The morphological operation of skeletonization can be used to capture topological properties of image objects [101, 89]. An extended object is eroded such that only a single-pixel-wide skeleton of its original shape remains. The skeleton preserves the number of connected objects, cavities, and holes in the original shape. For convex objects it can be used to determine the number of linked object (i.e. in clustered cells) and the relative angles between those objects.

2.5 Denoising and Deconvolution

Although the spatial resolution of a confocal microscope is much improved over the one of a conventional wide-field microscope, it is still subject to image degradation. While image degradations are commonly referred to as noise, one can distinguish four different sources [166]. 1) Statistical variations due to the Poisson statistics of quantum events and Gaussian distributed disturbances in the digital imaging system. 2) Scattering artefacts due to heterogeneous refractive indices within the specimen and index mismatching between immersion and specimen embedding medium. 3) Glare of lenses or filter in the imaging systems. 4) Blurring, which is caused by the light passage through the imaging system.

Since the statistics for the quantum events and the disturbances in the digital imaging system are known, they can be corrected with proper filters [166]. This is commonly done by image pre-processing steps in image processing pipelines. Scattering artefacts are sample specific and can not be predicted without prior knowledge of the different refractive indices and their distribution inside the specimen. However, refractive index mismatching between specimen embedding and objective immersion medium can be avoided by choosing suitable objectives and testing a range of immersion media. Lens or filter glare are prevented in commercial imaging systems by anti-reflective coatings which effectively prevent these imaging defects [166]. Similar to glare, blurring of the image due to diffraction limitations are instrument specific and specimen invariant.

The blur of an optical instrument is described by the point spread function (PSF) and can directly be measured with small fluorescent beads [30]. Beads with a size smaller than the diffraction limit are not captured as points but rather distorted by the optical instrument into 3D enlarged volumes. The fluorophores in a physical specimen are smaller than the diffraction limit as well. Thus, the captured image I_{Image} is produced by a convolution of the physically correct image

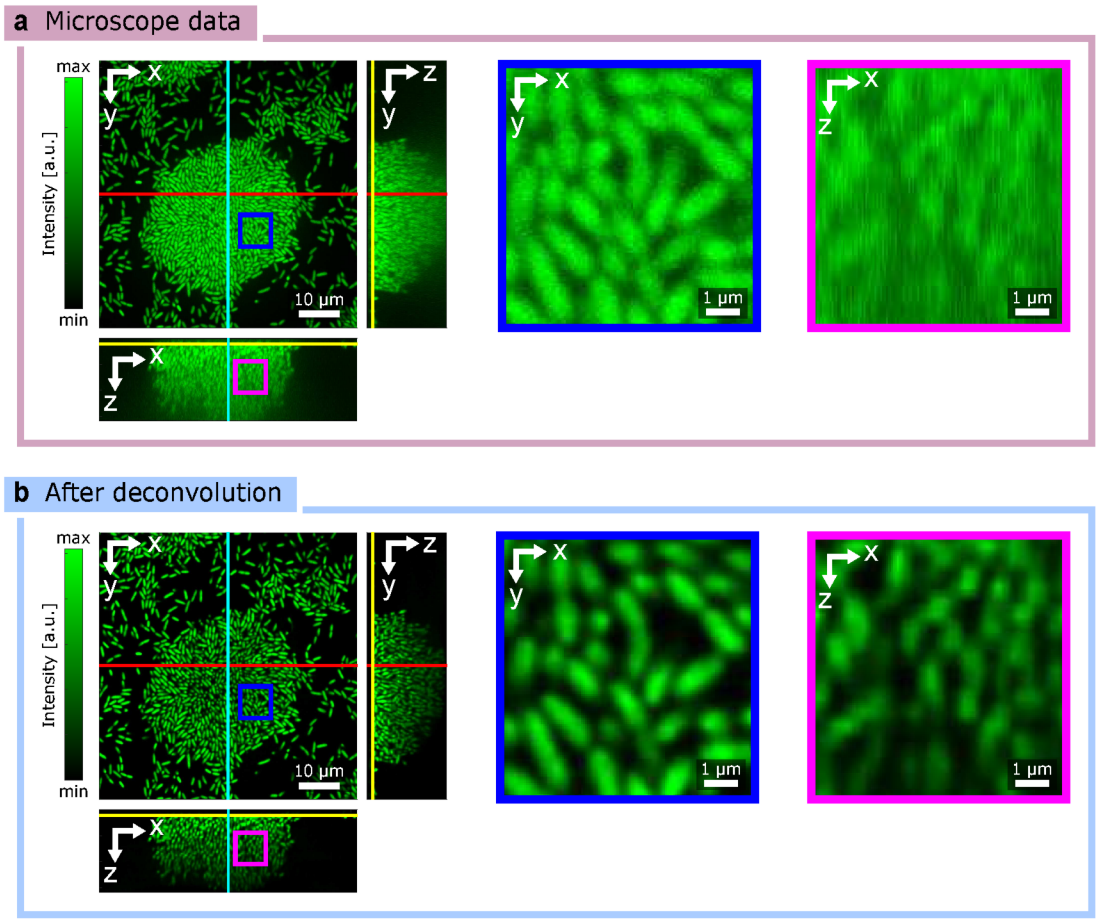


Figure 4: Example deconvolution on a three-dimensional (3D) confocal microscope stack of a *Pseudomonas aeruginosa* biofilm. **a** 3D sliced view of the biofilm for selected xy-, xz-, and yz-planes. Example regions in the xy-direction (blue) and xz-direction (magenta) are shown with 10× magnification. **b** 3D sliced view of the example volume after the blind deconvolution with *Huygens*.

of the specimen I_{specimen} with the PSF

$$I_{\text{Image}} = I_{\text{specimen}} \circledast PSF. \tag{17}$$

When the PSF is known, the inverse mathematical operation can be used to recover the physical image of the specimen up to the effects of the remaining image degradations [57]. This is called deconvolution and the influence on the image quality is shown in Fig. 4. The figure shows a captured *Pseudomonas aeruginosa* biofilm volume prior to the deconvolution (Fig. 4a) and after the corresponding operation (Fig. 4b).

Instead of using a measured PSF, the PSF can be guessed *a priori* by key figures of the experimental setup such as microscope type, used objective, immersion medium, and emission wavelength of the used fluorophore. This initial PSF is iteratively modified until the algorithm produces diminishing returns. This approach is called blind deconvolution and a series of different algorithms exist (i.e. Lucy-Richardson [137], Wiener [173], maximum likelihood estimation (MLE) [78]) for its application.

While blind deconvolution is computationally costly, deconvolution heavily affects the accuracy of the down-stream quantifications in the image analysis pipeline. Especially the image analysis on low SNR images benefit from a carefully conducted deconvolution. Recent progress indicates that the computational cost can be reduced and thus enable the tracking of single bacterial lineages inside bacterial biofilms with high temporal resolution [134, 61].

2.6 Segmentation

Image restoration and image enhancement are often required pre-processing steps for the feature extraction. For the feature extraction, however, individual cells need to be identified in the 2D image or 3D volume. The association of individual pixels (or voxels) to different object classes or individual objects is called segmentation. Only with segmentation cell phenotypes, shapes, sizes, or gene expressions indicated by fluorescent reporters can be extracted.

In multi-object images, computer vision distinguishes between semantic and instance segmentations [87]. Semantic segmentation associates pixels to different object classes. In the case of biofilm micrographs, the different classes are usually foreground (i.e. cells) and background (i.e. void). In multi-species biofilms, the foreground class can be sub-divided into the different observed species in the biofilm [183]. When a fluorescent microscope is used, a semantic segmentation into foreground and background pixels can be accomplished by determine an automatic threshold on the detected intensities and performing a binarization step [41, 179].

Instance segmentation tries to divide pixels sets into the shown objects. In the case of biofilm micrographs, each cell represents a separate object. In order to extract cell features from the objects, a single-cell segmentation is needed. To

produce a single-cell segmentation, all pixels which belong to an individual cell are assigned to the same unique identifier (ID).

Since segmentation algorithms are essential for the feature extraction in microscope images, a wide range of segmentation approaches exist [139, 41, 162, 7, 66, 167, 168, 183, 172, 156, 20]. The simplest algorithms for instance segmentation use a threshold based semantic segmentation and morphological operations to divide the cells into connected components [41]. For well-separated and bright cells this usually works remarkably well. This approach can be substituted with watershed-based post-processing to separate clustered objects [179, 66].

Due to the recent success on macroscopic scales [69], machine learning [7, 16] and in particular supervised deep learning approaches have seen a constant rise in popularity [183, 172, 156, 20]. Instance segmentation which employ learning strategies usually perform better than filter-based segmentation approaches [160].

2.7 Segmentation Accuracy

The evaluation of the segmentation accuracy relies on metric definitions. In computer vision the Jaccard index [80] is often used to measure the segmentation performance [62, 22, 77, 177]. It is given by

$$J(R, S) = \frac{|R \cap S|}{|R \cup S|}, \tag{18}$$

and depends on the set of pixels in the segmentation result S and the corresponding pixels in the set of the ground truth label R. The Jaccard index is often also called intersection over union (IoU) due to its geometric interpretation [77].

While providing good accuracy estimations for semantic segmentations and instance segmentations for macroscopic objects [54], it is not suitable for the evaluation of single-cell segmentation accuracies in bioimage analysis [22]. The per-image IoU does not capture missing and merged objects. Thus, metrics which count the object detections and penalize single object detection errors emerged.

To quantify individual segmentation errors, the IoU values between all overlapping objects in the segmentation result and ground truth labels are calculated. This results in a sparse matrix of IoU values. With the Hungarian method [96] an optimal matching between ground truth and segmentation results can be calculated such that the mean IoU value is maximized. Instead of using the global IoU value, a mean value can be calculated and used to quantify the segmentation accuracy. This already provides a usable metric to compare the segmentation accuracy between different segmentation algorithms [160].

As a rule of thumb, a segmentation with a mean IoU below 0.5 is indistinguishable from a false positive (FP) detection. A mean IoU value of 0.7 indicates a good segmentation result, a value of 0.8 indicates a very good segmentation, and a value of 0.9 is close to human annotation accuracy [48]. A mean IoU value heavily

penalizes segmentation algorithms which might work on some objects remarkable well, but fail to identify others. To detect such systematic segmentation problems, the matched IoU values can be used to define a metric which uses a user-defined IoU threshold τ for true positive (TP), FP, and false negative (FN) detections.

An object in the segmentation results is considered as FP if no matching object in the ground truth data was found with an IoU $> \tau$. A FN emerges, if for an object in the ground truth data no matching object in the segmentation with an IoU larger than the threshold exists. Consequently, a TP is detected if the matching objects exhibit an IoU value above the threshold. Based on these definitions the average precision [103, 172] can be defined as

$$AP = \frac{TP}{TP + FP + FN}. \tag{19}$$

A similar measurement, which puts more weight on the TP detections, is the F_1 score [22, 168, 183]:

$$F_1 = \frac{2TP}{2TP + FN + FP}. \tag{20}$$

2.8 Deep Learning

Deep learning is a subdomain of machine learning [56]. It uses deep artificial neural networks to transform an input signal into an output signal. Artificial neural networks are inspired by the connection of neurons in the human brain [98]. Similar to its biological counterpart, it consists of single neurons which process information via directed and weighted connections. In artificial neural networks, the neurons are usually organized in layers. If a large number of layers are applied sequentially, the emerging network is called deep artificial neural network.

In general, machine learning algorithms can be distinguished in two classes. (1) Algorithms which require a target value to be optimized (i.e. trained) and (2) algorithms which do not. While the first type of algorithms are called supervised learning algorithms, the second class are described as unsupervised [56]. Supervised deep learning is used to approximate an arbitrary function between input and target values during the training process [56]. While unsupervised methods have shown remarkable results in reinforcement learning tasks [117, 149], supervised methods produced astonishing results in cell segmentation [139, 81], breast cancer detection [114], and protein folding prediction [145].

The mathematical foundation of neural networks was established in the last century. The breakthrough, however, only emerged with the usage of networks with a large number of trainable weights [93]. The training (and application) of those networks, became feasible with the recent progress in compute resources [135]. Based on those improvements, the adaptation of the method in biological image analysis steadily increased [118]. The successful application of deep learning relies on a sequence of four distinct stages: Data preparation, training process,

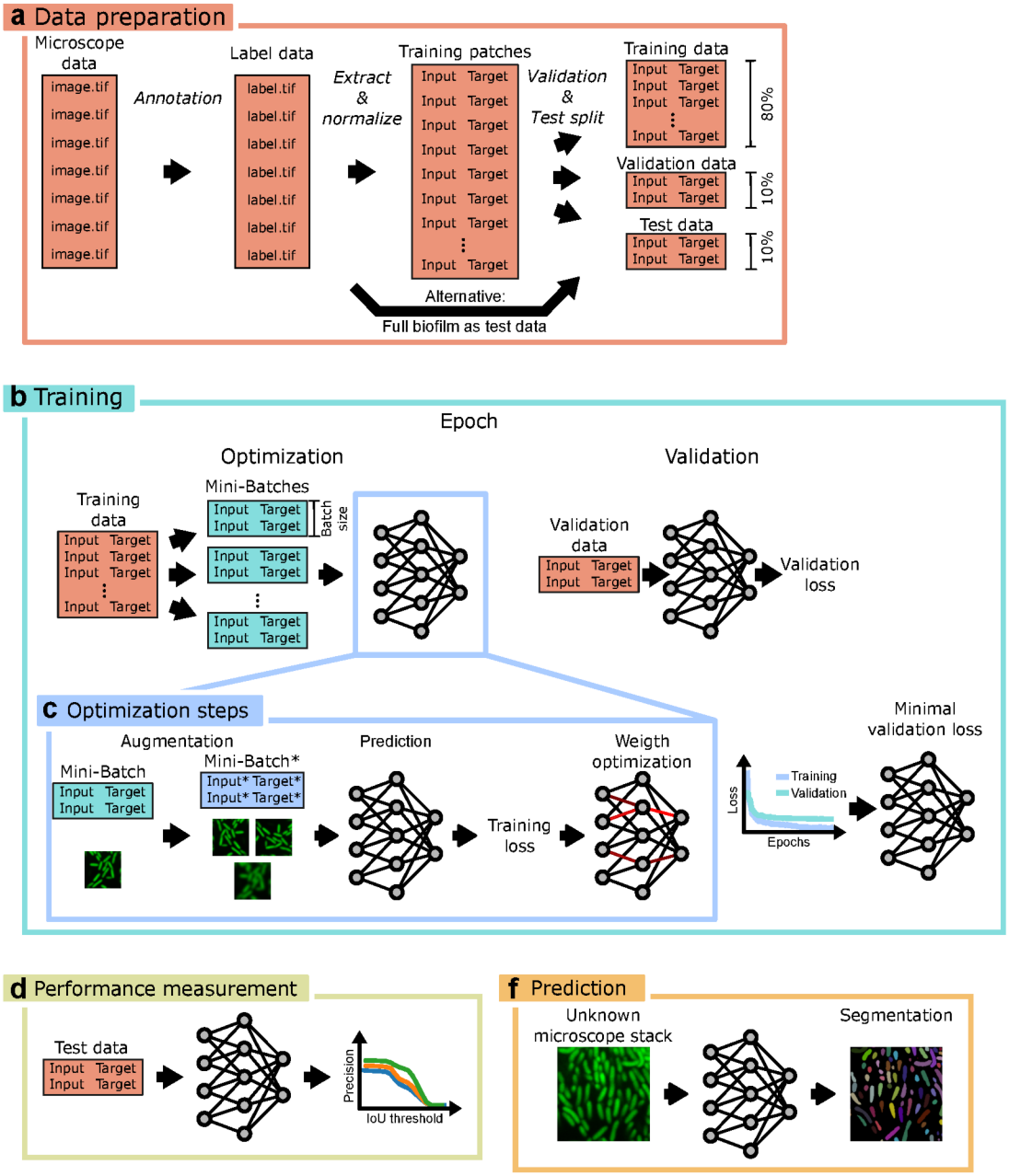


Figure 5: Schematic overview of the neural network training process. **a** Microscope data is gathered and annotated. Next, the image volumes are normalized and split into smaller volumes (patches) for the training process. Prior to the training process, the training patches are split into training, validation, and test data. **b** The training data is processed in several passes (epochs). In each epoch, the data is shuffled and split into mini-batches. Each mini-batch is processed by the network in optimization steps. A step can contain data augmentation prior to the feed-forward process. The weights of the network are optimized based on the training loss. Once all optimization steps are finished (i.e. at the end of an epoch), the validation dataset is used to assess the training performance. Usually the network with the minimal validation loss is used. **c** The performance is measured on a test data set which was not used for the weights optimization or the network selection. **d** The trained network can be used to segment data similar to the training data.

evaluating of neural network architectures, and the application on unannotated data (Fig. 5) [118].

Initially, the training data needs to be collected and prepared (Fig. 5a). In the case of supervised segmentation algorithms this also includes the annotation of a suitable large dataset. Suitable large means, that the input space is sufficiently sampled. After the annotation, training patches of input and target data are extracted and normalized such that at least a single training patch fits into the graphic processing unit (GPU) memory of the hardware. The normalization is necessary to make sure that during the training, the non-linearity in the network architecture is optimally used [68]. Finally, the training patches are divided into three distinct datasets: (1) The training dataset, which is used for the optimization of the network weights during the following training process. (2) The validation data, which is used to validate and select a suitable training epoch of the trained network, and (3) a test set, which is used to evaluate the performance in a realistic application scenario.

While the data preparation is the most time-consuming step, the training of neural networks requires the most compute resources (Fig. 5b). The training process is divided into multiple epochs. Each epoch is further sub-divided into individual optimization steps. In every epoch, the full training data is sampled in the optimization steps. Since the full training data usually does not fit into the GPU memory, the data is divided into so-called mini-batches [58]. The number of training patches inside a mini-batch is called batch size.

For each mini-batch an optimization step is started. To artificially increase the training data coverage of the input space, data augmentation techniques (i.e. rotations and flips along the symmetry axes, elastic deformations, intensity rescaling, or artificial noise insertions) are used [147]. These augmentations can either be applied prior to the training process (offline) or conducted during each step (online). The on-the-fly application comes with additional requirements on the compute hardware, but reduces the storage requirements of the training data by orders of magnitude [162]. For the optimization itself, the input patches are predicted by the network and compared with the target patches according to a loss function.

Based on the value of the chosen loss function, the weights w_i of the neural network at the optimization step i are optimized according to

$$\begin{aligned} w_{i+1} &= w_i + v_i \quad \text{with} \\ v_{i+1} &= m \cdot v_i - d \cdot \alpha \cdot w_i - \alpha \cdot \left\langle \frac{\partial L}{\partial w}|_{w_i} \right\rangle_{D_i} . \end{aligned} \tag{21}$$

Here, the momentum is given by v, the learning rate is denoted as α, and the weight decay as d. $\left\langle \frac{\partial L}{\partial w}|_{w_i} \right\rangle_{D_i}$ denotes the average gradient over the ith batch D_i of the derivative of the loss function L with respect to the weights w evaluated at w_i [93]. The optimization of neural networks with averaged gradients on randomly selected training patches is called stochastic gradient descent (SGD) [98]. The optimization steps are repeated until the full training data is processed.

At the end of the optimization epoch, the optimized network is validated. To prevent systematic errors, the validation dataset should be different from the used training data. The validation monitors the training process with respect to overfitting (i.e. the network reproduces 1:1 the training data [118]) and measures how well the network generalizes on previously unseen data. The result of the training process is an optimized neural network with a minimal validation loss.

In general, multiple networks with different architectures are trained. To extract the best performing network, a performance measurement is required (Fig. 5d). For the performance test, the network predicts the test data and the results are compared with the expected target value. This results in an unbiased measurement of the performance of different network architectures.

Finally, the optimized algorithm and best performing algorithm can be used to predict unlabelled microscope data (Fig. 5d). With well-optimized algorithms large amounts of data can be predicted.

3 BiofilmQ: A Tool for Quantitative Image Analysis

This chapter describes the capabilities of a GUI application that originated from a collection of *MatLab* (The Mathworks) scripts written and collated by R. Hartmann, H. Jeckel, and me. The scripts were conceived by R. Hartmann, which were used in previous publications [17, 51, 151, 164, 111, 39], and in particular [66]. Based on the unstructured script collection, R. Hartmann, H. Jeckel, and I created the stand-alone GUI software *BiofilmQ* [65].

I contributed feature extensions, several interface panels, and rewrote large parts of the code with respect to runtime, hardware requirements, and scientific rigour, created the *Windows* executables, wrote the online documentation[1], fixed an uncounted number of small and large errors in the code, and set up the required infrastructure necessary for maintaining an application software product of the presented scale. In addition, I conducted and analysed selected experiments, which underpin the applications of the described image cytometry approach in comparison to standard flow cytometry experiments, created figures for the corresponding paper, recorded online tutorials[2], and wrote the manuscript together with the other authors.

BiofilmQ is constantly adapted and modified to serve as the main analysis tool for all recent publications within the research group. Thus, in addition to the published version, which can be downloaded from the official website[3] as well as the *github* repository[4], an experimental version exists, which is reserved for internal use. This version includes custom-made features which are not included in the public version as they benefit single researchers with specific project requirements and require fine-tuning to produce valid results.

In the remainder of this work, I will describe the capabilities of the software and present the extended feature set which was developed for the research group. Hence, the term *BiofilmQ* will be used for the full software package. A special effort will be made to explain the features that are not included in the latest public release (version v0.2.1). For a comprehensive description of the published features, I would like to refer the reader to the *BiofilmQ* online documentation [84].

3.1 Introduction

Most interactions between microorganisms and the environment or between different microbes occur in microbial communities. Due to the self-produced matrix, microbial communities are resilient against external stresses such as tempera-

[1] `https://drescherlab.org/data/biofilmQ/docs/index.html`
[2] `https://www.youtube.com/playlist?list=PLzmiOL-opPEoo_Zyymcq0-WT4tevmiGhJ`
[3] `https://drescherlab.org/data/biofilmQ/docs/usage/installation.html`
[4] `https://github.com/knutdrescher/BiofilmQ`

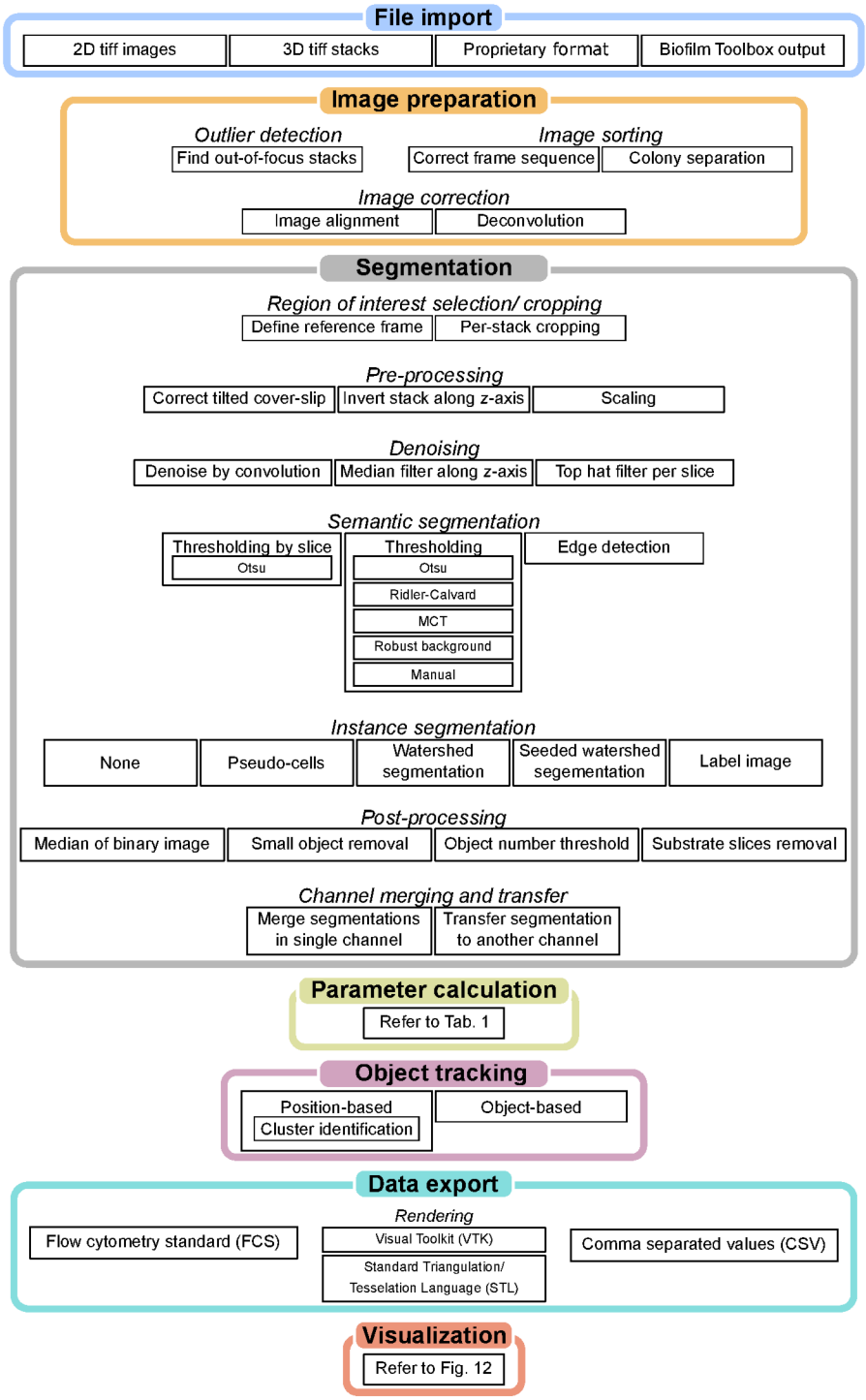


Figure 6: Overview of all analysis modules included in *BiofilmQ*. The workflow can be divided into seven steps. In each step, multiple options and parameter settings are available.

ture, media fluctuations, exposure to antibiotics, or predation by bacteriophages. Hence, the control of biofilm growth and dispersal is important in industries that are often challenged by the presence of biofilms such as agriculture and healthcare. For the full control of the processes that govern these multi-cellular aggregates, it is required to uncover and understand the signalling pathways inside biofilms and the precise response mechanism of biofilms to environmental changes.

Factors that influence biofilm growth and dispersal can be studied by growing biofilm-producing strains in laboratory conditions. Studying these factors relies on accurately measured experimental data. Nowadays, many biofilm studies include imaging using confocal microscopy, which is able to visualize spatio-temporal variations within biofilms due to developmental stages, exposure to stress factors, and cellular architecture [41, 151, 164, 66, 38]. While the data collection with confocal microscopy enables the observation of single cells inside the dense microbial communities, image analysis is essential to extract relevant biological features to test proposed hypotheses. Most published papers rely on bespoke image analysis, which are often not sufficiently described in publications due to the emphasis given on the characterization of a biological phenotype [12]. Without transparent analysis pipelines, it becomes impossible to compare and adapt results with the existing literature.

The reproducibility in the realm of (bio-)image analysis can be improved by shifting from custom-made pipelines to publicly available image analysis programs such as *COMSTAT* [73]. *COMSTAT* can be described as a first effort to make the quantification of phenotypic observations in biofilm experiments accessible to a broad user base. *COMSTAT* and its successor *COMSTAT2* [165] received wide acceptance in the biofilm research community. However, since its release in 2000, the biofilm research shifted from observations on the range of full microcolonies to the description of processes at the single-cell level. Yet, the implemented feature extractions have not evolved. The limitation of *COMSTAT2* becomes apparent when heterogeneity in different spatial regions of a biofilm is investigated.

This spatial heterogeneity is driven by variations in the nutrient and oxygen availability, as well as the diffusion of metabolic waste products within the biofilm. In contrast to cells in the interior of the biofilm, cells on the exterior are directly exposed to a constant flow of new nutrients and oxygen from their surrounding medium. Cells in the inside, however, lack the direct nutrient access and therefore, exhibit a different metabolism. *Bacillus subtils* biofilms show growth rate oscillations between cells located in the periphery and in the interior of the colony in order to prevent starvation of the interior cells [105]. To capture such spatio-temporal changes during the development of large biofilms, large amounts of image data are required. Without automatized image pipelines, the analysis quickly becomes infeasible for time-lapse experiments." The metabolic difference between the cells on the interior and the exterior of a colony, also manifests in the shape of different morphological properties of single cells [83]. To capture these differences, not only spatially-sensitive image analysis, but also single-cell segmentation is required. Single-cell segmentation can enable the modelling of physical interactions inside the biofilm and its influence on the overall biofilm

morphology [41, 66].

Linking cellular genetic changes with morphological effects in biofilms is important to explain the intrinsic factors responsible for a biological phenotype [179, 38]. However, the required image analysis algorithms are out of reach for many microbiology laboratories. *Fiji* offers a toolbox for setting up such a pipeline without re-implementing low-level functions [142]. Yet, it sacrificed flexibility for intuitive usage, such that the user gets lost in the many available options, only some of which are relevant for biofilm investigations. Furthermore, its batch processing capabilities require the recording and modification of macros. Learning how to modify these *Fiji*-specific macros is comparable to the usage of a high-level programming language.

To overcome the limitations of image analysis programs such as *COMSTAT* and *Fiji*, we developed a tool specialized for biofilm research called *BiofilmQ*. It is designed to enable its users to extract physical properties from fluorescent image data with spatio-temporal resolution. Applications include the spatio-temporal description of fluorescent reporter levels inside dense communities [151], the quantification of the penetration depth of phages, fluorescent beads or invading cells in biofilms [164, 38], and the spatial distribution of biofilm strains in multi-species communities or host organelles [53]. It combines the image analysis experience of numerous publications into a simple GUI tool in order to accelerate the time spent on analysis. For image analysts, it offers the possibility to quickly implement feature extensions in new modules. The analyst can focus on the feature extraction, while routine tasks (i.e. file handling, denoising, segmentation, and visualization) are already covered by the program itself. Due to the integration into the existing GUI, end-users do not have to adapt their overall analysis pipeline every time a new feature extraction is implemented. Additionally, it comes with batch processing capabilities to scale the analysis quickly up and provides useful features for 3D rendering, file import, 3D viewer and more.

3.2 Workflow Overview

BiofilmQ is grouped into seven analysis steps of file import, image preparation, segmentation, feature extraction, cell tracking, data export, and visualization (i.e. plotting). Each step comes with different modules which provide the required flexibility to operate in a divers research environment with different analysis requirements. In Fig. 6, all available modules of *BiofilmQ* are listed. In practise, not all modules might be required to extract the desired features from the image data. Pooling them in a single GUI application, however, enables the user to quickly pivot the analysis approach if needed.

The analysis steps are ordered such that they come together in an intuitive workflow (Fig. 7). By organizing the steps in different tabs, the user performs the task in a sequential order and can skip functions of *BiofilmQ* if they are not required for the analysis goal. With this simple design, the user is guided through all necessary steps which lead from data import to the visualization of the extracted

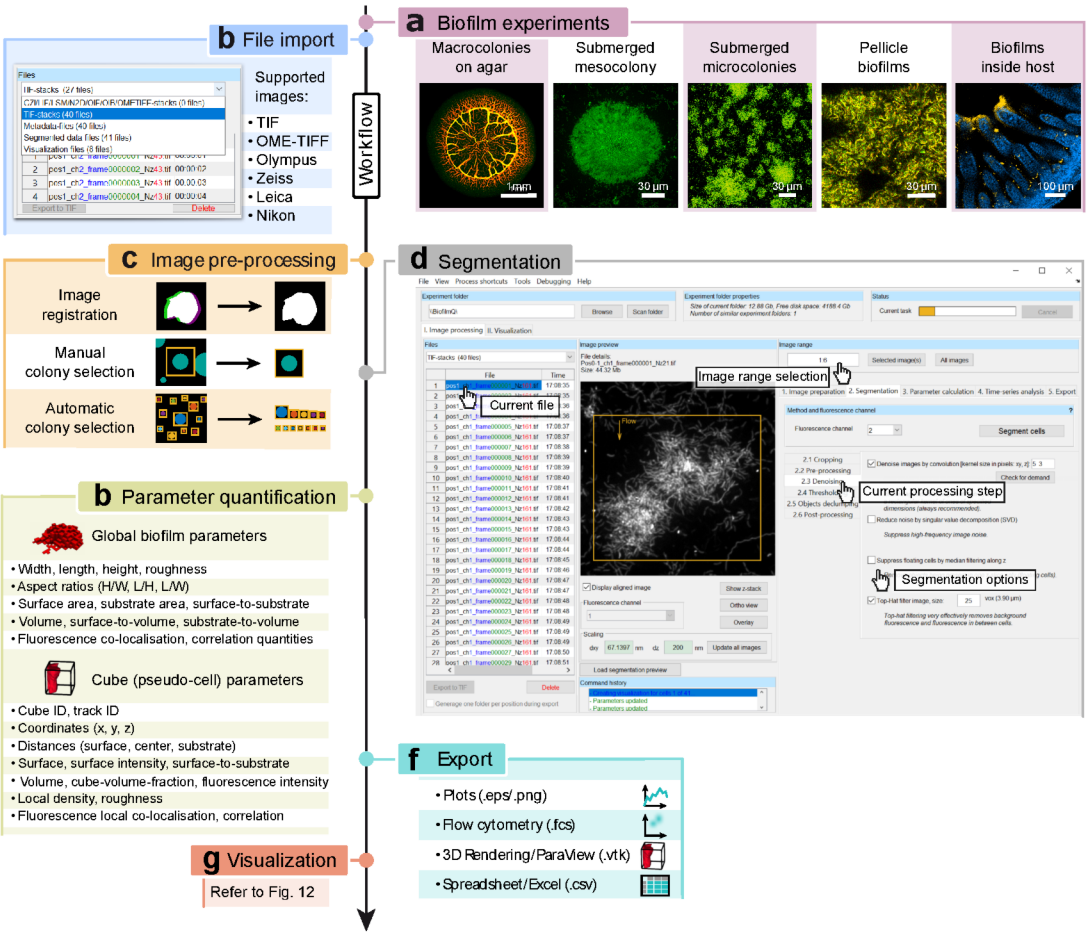


Figure 7: Example workflow within *BiofilmQ*. **a** The biofilm is imaged in one of many possible experimental settings. **b** The collected image data in one of the supported formats is imported into the analysis workflow. **c** Pre-processing steps such as image registration and colony selection is applied. **d** The image is segmented with the graphical user interface (GUI). **e** Based on the segmentation, features of the objects (here: pseudo-cells) are extracted. **f** The features are exported to standard file formats for sharing or importing in other analysis tools. **g** Based on the extracted features, multiple visualization options are provided. Figure modified from [65].

features without leaving the GUI.

The application was developed to suit a variety of biofilm experiments (Fig. 7a). The size of bacterial biofilms can range multiple orders of magnitude. From visible macro-colonies, which typically grow on agar-based medium in Petri dishes in the range of several millimetres, to submerged colonies in flow chamber experiments, pellicle biofilms at liquid-air-interfaces, up to experiments in host systems like intestinal tissue inside young mice [53]. The only requirement for the analysis with *BiofilmQ* is availability of a fluorescent signal to detect the biovolume in the captured images.

3.3 File Import

During experiments, image data is collected either with a confocal fluorescence microscope or on an epifluorescence microscope in the corresponding image capturing software. This software can be either proprietary software which is maintained by the microscope vendor (i.e. *NIS Elements*[5]) or open-source microscope control libraries (i.e. *μManager* [45]). Based on the microscope vendor and the experimental setup, the user has to import the microscope file formats (Fig. 7b). To support all major vendors, *BiofilmQ* relies on the *Bioformats MatLab* bindings [104].

For each experiment, most proprietary data formats compress the acquired image data into a single file. This results in files which not only contain image stacks from several time frames or acquisition channels, but also from different positions inside the specimen. *BiofilmQ* offers the option to extract the image stacks either into a single folder, or to create separate folders for each position. The former (the extraction into a single folder) is advantageous to create demographs for a single time-frame (i.e. to compare different strains with each other), the later (extraction into several folders) is beneficial for the analysis of temporal changes in time-lapse experiments.

For the import of tagged image file format (TIFF) files, a custom import routine was written which scans the directory, detects all TIFF files and offers different import options based on the filename and data type. If the filename indicates that the files are already in a *BiofilmQ*-compliant format (i.e. frame number is given by a six-digit number with leading zeros after the characters "*_frame*" and the number of z-slices is given after the prefix "*_Nz*"), the files can directly be used in the *BiofilmQ* application. Otherwise, an import view appears.

In the import view, the user can specify prefixes to read metadata (i.e. ID of the acquisition channel, the stack ID, and the frame of time-lapse experiment). In the case of 2D TIFF files, the ID of the z-level in which the slice was taken is required. The program tests all file names in the current folder with respect to the user-defined prefixes and presents the results in a table. For the reconstruction of 3D volumes from imported 2D images, the user has to specify the z-position inside a 3D volume to generate the stack. If the required metadata is not encoded in the file name, or the pattern matching could not detect a given pattern, the program asks the user to manually insert the metadata either by giving a default value or by editing the metadata for each file individually.

Based on the metadata, the software creates a folder structure and changes the file names to fulfil the previously stated format conventions. Each position is saved in a separate folder, which contains the image data, the image metadata files, and a parameter file which stores the GUI settings of the analysis steps for documentation purposes.

[5] `https://www.microscope.healthcare.nikon.com/de_EU/products/software/nis-elements`

3.4 Image Preparation

Prior to the image segmentation, the image data can be investigated for experimental artefacts such as lost focus positions, sample drift, or wrongly labelled frame numbers (Fig. 7c). Additional functions include the possibility to create batch files for image deconvolution, or working with multiple regions of interest (ROIs) in a single time series.

Out-of-focus stacks

Even though most commercial microscopes are equipped with an active autofocus system (i.e. *Perfect Focus* for *Nikon* microscopes), focus drift can occur during imaging sessions [85]. These focusing systems work by detecting differences in the refractive index along the z-axis of the specimen. Refractive index differences, however, do not only occur between glass substrate and bacterial population but also exist between bacterial population and (water-based) culturing medium [9]. Such focusing errors result in image stacks, which do not contain any biofilm-bound cell. For the analysis of time-series experiments, it is beneficial to remove these empty (i.e. out-of-focus) frames from the dataset and rely on nearby data points to display the dynamics in the time-lapse experiment.

To prevent manual user intervention for a significant number of out-of-focus frames, *BiofilmQ* includes a semi-automatic detection and deletion mechanism. It identifies the substrate plane and calculates the mean of the brightest 200 pixels in the plane above the substrate. The values are plotted in a diagram and the user can define a threshold for the whole experiment. With the described, heuristic out-of-focus score, the corresponding frames can be robustly identified. All volumes with a score below the user-defined threshold are deleted after a notification.

Frame numbering correction

When used improperly, custom microscope control software i.e. the ones used by Hartmann et al. [66] can produce overlapping frame numbers. A possible reason is a camera crash, followed by a restart of the control software. The user ends up with multiple volumes which have the same frame number, yet are labelled with different time stamps. The frame numbers can be corrected by relabelling the TIFF files in the experiment folder according to the time stamp.

Deconvolution

To improve the axial resolution in microscope images, deconvolution has become the method of choice to recover the physical image. Blind-deconvolutions based on MLE are a powerful tool to improve the SNR in microscope images [15]. It is

to be noted, however, that besides their ease of use and immediate improvement of the SNR, they are still inferior to experimentally recovered PSF and system-specific noise models. A commonly used application is *Huygens*[6]. To simplify its usage further, the GUI includes the option to create a batch file bespoke to typical microscope settings in the research group. Additionally, the user can select the excitation as well as emission wavelength of standard fluorophores or set them to user-defined values. For the deconvolution process, the image files are first prepared for the use in the third-party deconvolution software, and – together with the batch file – copied to a shared folder on a designated high-performance server. Once the deconvolution is finished, the deconvolved images are reimported into the analysis pipeline. The results are directly available in an extra channel, such that the original image intensity information remains untouched. This allows the user to deconvolve large quantities of image data without manual interactions.

Colony separation

When working with small magnification lenses, multiple biofilms can emerge in a single captured volume. However, the currently implemented segmentation pipeline only works with an single biofilm per stack. If the user wants to analyse multiple ROIs, a copy of the image data has to be created and – depending on the desired analysis – stored in a separate folder. If the user wants to create a demograph of the biofilms in one field of view at a single time frame, the data is best duplicated in the same folder with a different ROI definition. To automatize the ROI definitions, the mean of the 3D volume along the z-axis is calculated and illustrated in an overview image. With the selection tool, the major structures can be marked. The selection is based on the pixel intensity of the objects in the mean projection and the object size. By using a higher size threshold, smaller objects – such as single cells – are filtered out. After the object selection, the folder structure is updated, such that the new stacks can be used in separate downstream analysis steps.

Image alignment

For the calculation of spatio-temporal biofilm or single-cell parameter, the measurement reference has to be fixed with respect to the biological sample. During long time-lapse microscope experiments however, a focus drift can often be observed [85]. While the microscope vendors typically provide a technical solutions for the focus drift, sample drift in the xy-plane is not corrected.

To mitigate this drift phenomenon, *BiofilmQ* contains a robust alignment routine which uses spatial correlation or mean square displacement error (MSDE) to correct the sample position during time-lapse experiments. The spatial correlation based method uses the discrete Fourier transformation to achieve sub-pixel

[6]Huygens Essentials version 19.04 (Scientific Volume Imaging, The Netherlands, http://svi.nl).

volume registration [60]. It offers a good realignment for small displacements and works with images of different intensity levels and varying noise. The MSDE algorithm, however, can be used for larger displacements but works best on images with similar brightness and contrast.

Since both alignment algorithms are only available for 2D image data, *BiofilmQ* uses 2D projections to estimate the displacement of volumetric data. Here, the mean value along the z-axis is projected onto the xy-plane. For the projection onto the xz- and yz-plane, the sum along the y- and x-axis is calculated, respectively. In a next step, the 2D translations between the projections of the current image stack and the projections of its predecessor are extracted. The final translation is the sum of the translation in the xy-plane and the mean value of the z-displacement in the xz- and yz-plane.

3.5 Segmentation

The core element of any object-based image analysis pipeline is the image segmentation (Fig. 7d). In *BiofilmQ*, the image segmentation pipeline is fully controlled via the GUI in which the user can select files for the analysis or for preview, set the range for the image analysis, and set different parameter for the segmentation steps. The segmentation can be split into different processing steps.

Region of interest definition

To capture the full specimen, the imaged volume is usually larger than the specimen itself. This results in a large amount of voxels which do not contain any information on the specimen. In this context, voxels directly translate to data which has to be processed, stored, and read by the software. More data results in longer processing (and input/ output) times and can significantly slow down the analysis pipeline. Thus, it is advisable to select only a small fraction of the specimen to quickly test new analysis parameter. Both can be done by defining a suitable ROI. *BiofilmQ* supports two types of ROI definitions: The analysis ROI and the so-called reference frame. The analysis ROI defines the extent of the volume which needs to be analysed. The reference frame further defines a stationary region which is used to measure the spatial coordinates of the experiment and benefits the comparison of absolute spatial properties such as centroid positions or voxel coordinates. Additionally, a fixed reference frame is beneficial for visualizations which need a fixed output size such as time-lapse films of biofilm growth. While the reference frame has to be defined once per time series. The ROI is defined for each frame separately. To simplify the ROI definition, *BiofilmQ* includes an interpolation tool such that the user only defines the ROI in key frames and the tool calculates a linear interpolation of the ROI for all remaining frames.

Image Preprocessing

Based on the experimental set up, biofilms can be imaged from above or from below. Due to the available inverted confocal microscopes in the research group, *BiofilmQ* is optimized for bacterial cells which are imaged through a glass substrate. To account for up-right microscopes, the user has to transform the dataset by enabling a simple check box.

In some experiments, the normal vector of the substrate is not parallel to the z-axis, but rather tilted in space. To process these experiments nonetheless, the user can use a correction algorithm which estimates the position of the substrate plane and conducts a rotation of the full volume such that the plane normal direction is perfectly aligned with the z-axis of the image stack.

If necessary, scaling can benefit the image analysis. Especially large datasets quickly can exceed the available random access memory (RAM) and pose serious obstacles for the image analysis pipeline. By downscaling the image data by a user-defined factor, *BiofilmQ* can be used for these large datasets at the expense of reduced spatial resolution. Besides optimizing the spatial dimensions of the volume, the accuracy of segmentation pipelines benefits from modifications in the signal intensity dimension [20]. One challenge for segmentation pipelines is image noise which alters the SNR available for the algorithm [183]. The two major sources for intrinsic image noise are the variance in the Poisson-distributed fluorophore activity and Gaussian-distributed camera readout noise [166]. The camera noise can be reduced by cooling the camera (i.e. via Peltier elements) [127]. In general, a simple mean filter (9) pools neighbouring pixels together and thereby reduces the fluctuations in the individual pixel wells.

Another source of inconsistent intensity information emerges due to free-floating cells. Even with low camera imaging intervals fast-moving cells are often imaged in a single z-plane due to the acquisition delay between the different z-planes. These cells – although not contributing to the bounded biomass inside a biofilm – can introduce unwanted segmentation artefacts. Therefore, the user can remove these cells by applying a median filter (10) in the axial direction. The influence of floating cells is significantly reduced and segmentation artefacts can be prevented.

Even though high fluorophore concentrations are only expected inside the cell membrane, a stray fluorescent signal can be observed in the empty space between the cells. This signal can be introduced by the PSF of the microscope, different refractive indices due to densely packed cells, or pinhole crosstalk. The superior approach to eliminate intra-cellular noise would be via deconvolution (compare Sec. 3.4), another simple – yet effective – approach to reduce the unwanted signal is the application of the top hat transformation (16) [163]. In the current implementation, the top hat transformation is applied plane-by-plane to account for the signal intensity decay in deeper layers of the specimen.

Semantic Segmentation

BiofilmQ includes three semantic segmentation approaches, each relying on different algorithms. As explained in the introduction (Sec. 2.6), the semantic segmentation describes the process of assigning each pixel to an object class in contrast to an instance segmentation which identifies individual objects in the image.

Thresholding A very basic approach for a foreground/ background segmentation of fluorescence microscopy images is the application of a global threshold value. Due to their simplicity, the algorithms belong to the fastest available segmentation approaches [163]. All values below the threshold are assigned to the background, all voxels with an intensity above the threshold are assigned to the foreground (i.e. cell signal). In contrast to the user-defined thresholds, automatic threshold algorithms estimate values without user input. These algorithms have the advantage that they can be directly applied to images of varying intensities [163]. Besides the very versatile Otsu thresholding approach (Sec. 2.4) [128], *BiofilmQ* includes the Ridler-Calvard algorithm [138] which iteratively determines thresholds k_{i+1}. The iteration step from k_i to k_{i+1} is given by

$$k_{i+1} = \frac{\mu_0(k_i) + \mu_1(k_i)}{2}. \tag{22}$$

The iteration stops when the condition

$$k_{i+1} - k_i \leq \epsilon, \tag{23}$$

is fulfilled. Here, μ_y denotes the mean of the classes $y \in \{0, 1\}$. In the chosen implementation, k_0 is set to the mean intensity of the image, and ϵ is set to 10^{-6}.

Another included automatic thresholding algorithm is called maximum correlation thresholding (MCT) [130]. For a given integer image f it calculates correlation coefficients C for every possible threshold value k via

$$C(x,k) = \frac{\sum_{j=1}^{m} n_j \cdot (j - \bar{x})}{\sqrt{\left(\sum_j^m (j-\bar{x})^2 \cdot n_j\right) \frac{(N_m - N_k)\cdot N_k d}{N_m}}}. \tag{24}$$

Here, N_m denotes the total number of pixels values, n_j the number of pixels with the value j, $\bar{x}$ the mean of all pixels in the image, N_t the number of pixels above $(k-1)$, and m is the number of different pixel values in the image. The optimal threshold is given by the value k^* which maximizes C.

Additionally, *BiofilmQ* includes a robust background threshold implementation. The robust background algorithm uses all values larger than the positive 2σ values of a Gaussian approximation within the percentile rank 5-95 of the intensity values.

Finally, *BiofilmQ* comes with a manual threshold selection option, which includes a live feedback in a sliced 3D view. The viewer presents a xy-, xz-, and yz-plane for a user-defined position inside the specimen. Thus, the user can directly observe the influence of the selected threshold on the segmentation result.

Thresholding by Slice Due to non-optimal optical properties inside dense biofilm colonies, the fluorescent signal is heavily scattered. This results in a weakening of the fluorescent signal the higher the focal plane is located above the substrate [41, 179]. Therefore, the simple thresholding algorithms are complemented by a z-dependent implementation of Otsu's method. To avoid the detection of foreground voxels in empty z-planes, a variance score is calculated. If the variance score is above a certain threshold, the slice is segmented. Otherwise, the slice does not contain any foreground signal and is skipped.

Edge detection Semantic segmentations with thresholds may fail if the objects within the ROI exhibit difference brightness or if an inhomogeneous background is present [37]. In these cases, the Marr-Hildreth edge detection (13) offers an elegant solution for the segmentation problem. This method produces state-of-the-art results in the segmentation of single cells inside bacterial populations, especially when combined with a subsequent watershed segmentation to derive a mask for single cells [66, 10]. For an in-detail description of the chosen implementation, I would like to refer the reader to Sec. 4.

Instance Segmentation

In contrast to the assignment of each pixel to a certain object class, instance segmentations try to identify individual objects inside an image independently of the corresponding object class [118]. Thereby, instance segmentation enables the investigation of single-cell properties inside bacterial biofilms [41, 179, 66].

Pseudo-cells Although powerful imaging techniques for single-cell imaging inside bacterial biofilms such as dual-view light-sheet microscopy [25] or SDCM exist, not all studies might benefit from such sophisticated microscopy setups. The spatio-temporal measurement of fluorescent reporter activity inside different biofilm regions, for instance, does not require single-cell resolution. In some cases, e.g. *Bacillus subtilis* biofilms growing on lysogeny broth (LB) agar plates, it is impossible to segment single cells in the specimen due to inhomogeneous refractive indices inside of the dense community. Another problem is the limited working distance of high NA objectives. Cells on the agar surface are out of reach for the microscope focus. The imaging with an upright microscope, however, can not use objectives with immersion media, since it would alter the biofilm structure. Thus, microscopes are limited to low NA air objectives. Even if the agar plate is thin enough, the scattering of the signal due to the different refractive indices inside the specimen prevents single-cell imaging. Especially if no morphological properties of the cells (such as size, length of medial axis, or curvature) are required, the use of "pseudo-cells" provides a powerful approach for the spatio-temporal investigation of expression patterns, or density concentrations inside biofilms [17, 39]. Instead of investigating single cells, the biofilm is split into small cube-like objects which are assigned to each position inside the biofilm.

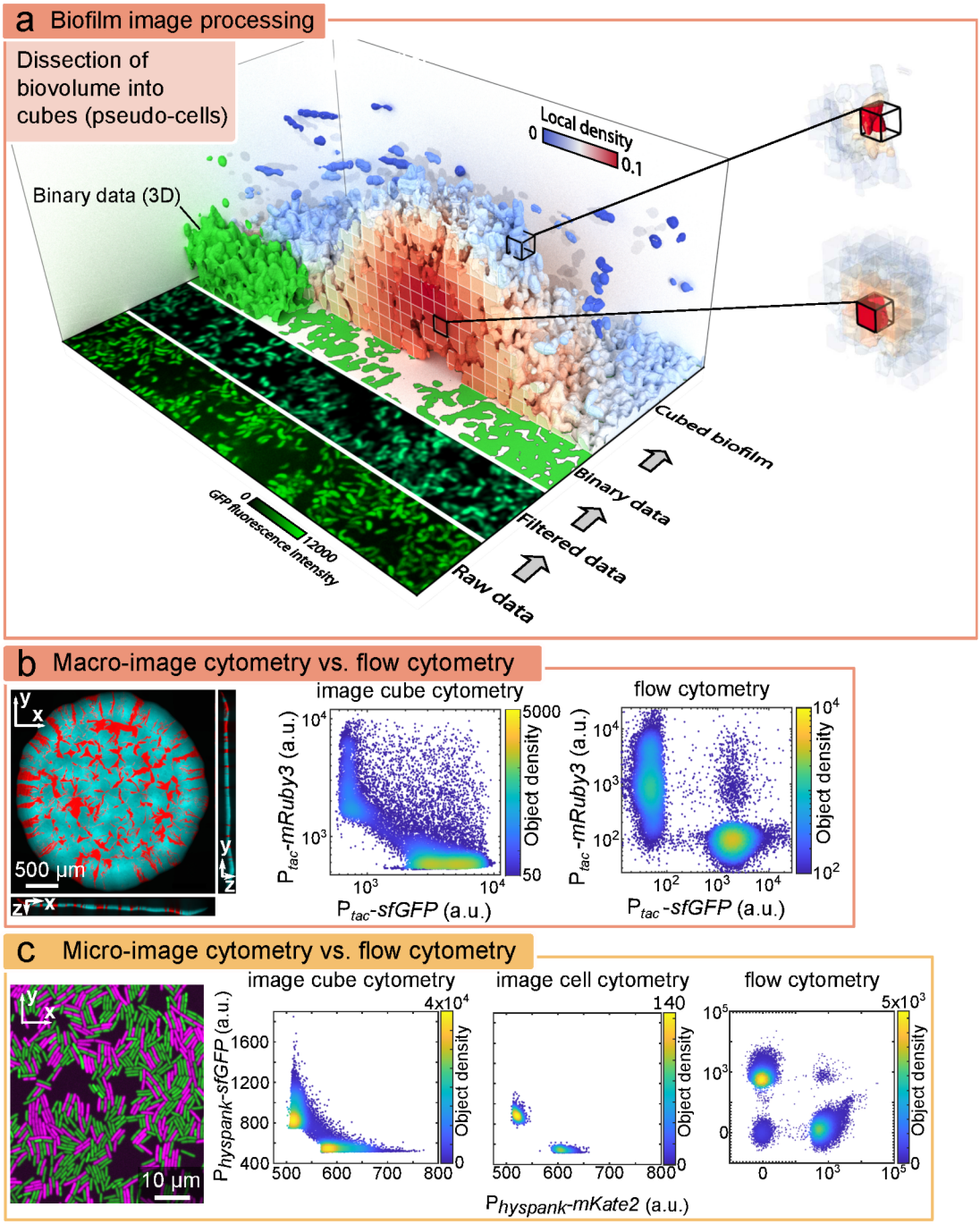


Figure 8: Instance segmentation with pseudo-cells. **a** Three-dimensional (3D) binary data is extracted via image processing pipeline and dissected into pseudo-cells (cubes). Each cube contains a fraction of the full biovolume and can be used for feature extraction (i.e. local density quantification). **b** Comparison between the proposed method and the single-cell flow cytometry analysis in an experiment with two *E. coli* strains co-cultured on agar. Left, grown colony consisting of two strains. Centre, results of the pseudo-cell cytometry analysis. Right, result of the corresponding flow cytometry analysis. **c** Comparison of pseudo-cell analysis on an image stack with single-cell resolution. Left, mono-layer of two *Bacillus subtilis* strains. Centre left & right, results of image cube cytometry and (single-)cell cytometry analysis. Right, result of flow cytometry analysis, respectively. Figure modified from [65].

The basic idea of using cubes as a pseudo-cell proxy is depicted in Fig. 8. The segmentation starts with the filtering of the raw image. During this step, the previously described filters are applied to reduce the influence on the segmentation results (Sec. 3.4). Starting from the filtered image, a binary mask is generated and divided into equally-spaced cubes. Based on the cube dissection, spatial properties can be calculated and used for the following analysis.

This method compares favourably with established single-cell cytometry approaches such as flow cytometry on macroscopic (Fig. 8b) and microscopic levels (Fig. 8c). To compare the image cube cytometry and flow cytometry results in a macro colony experiment, a mixed culture of two *Escherichia coli* strains was investigated. Each strain constitutively produced fluorescent proteins at different excitation and emission wavelengths. After the mixing, the inoculum was spotted on LB agar medium and incubated until the shown 3D macro colony was grown (left). The colony was imaged with two different objectives to a) capture the full extent of the colony and b) provide sufficient resolution to discriminate the two strains in a pseudo-cell segmentation. After the (virtual) dissection into cubes, the mean intensities of the single cubes in both channels were measured (centre). Here, the colour indicates the object density in the scatter plot. Directly after imaging with the confocal microscope, the shown colony was carefully removed from the agar surface and re-suspended in phosphate-buffered saline (PBS) solution, disrupted with a vortex shaker, and analysed with a flow cytometer (right).

In the image cube cytometry analysis, two populations with orthogonal intensity profiles are visible. The result qualitatively resembles the result of the flow cytometer analysis. Pronounced regions which exhibit only a single fluorescent channel are visible in both scatter plots. However, a quantitative overlap in object density and fluorescent intensity cannot be observed. The mismatch can be explained with the different excitation lasers and different emission filters used flow cytometer and confocal microscope. While the microscope laser and filter are carefully selected for the used fluorophores, the flow cytometer supports various fluorophores with the trait-off of non-optimal excitation and detection efficiency. Nevertheless, this proves the validity of an image cytometry approach for the investigated system.

In a second experiment, the differences in the fluorescent intensity on single-cell level (Fig. 8c) with the proposed cube cytometry, a single-cell segmentation, and a standard flow cytometry approach were investigated. For this experiment, two *B. subtilis* strains with constitutive fluorescent protein production were used. After growing them in liquid culture, the concentration was enhanced by centrifugation and the strains were mixed. The mixed culture was inoculated in a small droplet on a cover slip. The droplet was covered with a piece of LB agar to force the planktonic cells into a single-cell layer. A 3D volume stack of the layer was taken with a high NA objective (Olympus 100x 1.35 NA) (Fig. 8c, left). After imaging the 3D volume with the confocal microscope, the same planktonic culture was analysed with the flow cytometer.

The close relation between the image cube cytometry (Fig. 8c, centre left) and the single-cell cytometry (Fig. 8c, centre right) is visible. The intensities as well as

the regions with the highest object densities match almost perfectly. The major difference between the two approaches is the long tail in the image cube cytometry plot and the larger number of the plotted objects. For the data analysis, the cube size has to be smaller than the object size of interest. Otherwise, edge effects dominate the analysis and differences between the two populations disappear. This is also the reason why some objects in the image cube cytometry plot seem to exhibit two fluorophores at once. Those data points emerge from cubes which contain parts of cells of both sub-populations. The same effect can explain the long tail of higher intensities. The pseudo-cells in the centre of a fluorescent cell only contain pixels with a high intensity of fluorescent signal. In contrast, the full cell volume also contains dimmer regions close to the cell membrane and therefore exhibits a smaller mean fluorescence.

Additional to the two major populations with orthogonal fluorescent reporter activity, the flow cytometer analysis reveals one sub-population with no fluorescent activity and one population with activity in both fluorescent channels (right). The additional non-fluorescent population emerges due to cell debris in the overnight culture. Due to its missing fluorescence, this debris cannot be analysed with *BiofilmQ*. The double fluorescent signal however, is probably an artefact due to multi-strain doublets of cells. If these doublets pass the flow cytometer detector simultaneously, a signal in both channel is recorded. Doublet gating could have reduce these counts, but would introduce an additional bias in the data analaysis.

The presented panels Fig. 8b & c illustrate the interchangeable usage of single-cell, cube, and flow cytometry analysis. In contrast to the flow cytometer analysis, an image-based cytometry approach does not disrupt the biofilm during imaging and thus preserves the spatio-temporal information of the reporter activity. This enables investigations of reporter activity in dense biofilm populations.

Watershedding The idea of using the watershed segmentation to dissect small cell clusters into single cells, is a well-established approach in single-cell segmentation [179, 10, 66]. The watershed segmentation uses the greyscale image intensity as a topological map [8]. In this topological interpretation of the image, the bright centres of the cells become valleys, background regions become plateaux, and membranes of touching cells become ridges. If these (imaginatively) valleys are flooded with water and barriers prevent the rising water from crossing the ridges, compartments emerge. Each compartment comprises the complete volume of a single cell. This is not restricted to 2D images, but can also be applied to 3D greyscale volumes. For a detailed explanation of the implemented algorithm the reader is referred to Sec. 4.3 in which the combination of edge-detection for a semantic segmentation and a watershed-based dissection for single-cell segmentation is discussed.

Label Image In the past years, the field of biological image segmentation has experienced steady progress in segmentation accuracy and generalization capabilities of the published algorithms. Especially with the recent emergence of deep

learning [162, 48, 172, 156] and random forest-based [16] segmentation tools, project-specific segmentation pipelines soon will become obsolete. The current efforts of creating domain-specific segmentation approaches will be replaced by adapting the available algorithms to the project-specific segmentation task. The programming of bespoke segmentation algorithms will be replaced by the generation of suitable training data. With this development in mind, it becomes more and more important that the analysis workflow remains flexible enough to import the results of third-party segmentation tools into *BiofilmQ*. This enables the user to a) quickly switch to new algorithms, b) reuse the rich set of feature extractions and visualization capabilities provided by *BiofilmQ* itself, and c) benefit from the future development in segmentation algorithms which might produce better results for single-cell segmentations for a particular experiment.

To use the segmentation results of third-party tools, the user needs to create 3D TIFF stacks i.e. with the already presented import functions of *Biofilm* (compare Sec. 3.3) out of the segmentation results. After the modification of the file names, the user can use the implemented *Label image* dissection method to generate *BiofilmQ* compatible single-cell segmentations from the imported datasets.

Post-Processing

The post-processing after the segmentation can be divided into two parts. The first part removes artefacts emerging due to the used segmentation algorithms. The second part deals with the transfer or merging of segmentation information in order to process multi-channel fluorescent signals i.e. with differently fluorescently labelled reporter strains in a single experiment.

For the artefact removal, *BiofilmQ* comprises three help functions: 3D median filtering, removal of small voxel clusters, and removal of bottom slices. 1) The 3D median filtering of the binary image removes sharp edges in the object boundaries. These edges can emerge due to the pixel-wise semantic segmentation. In extreme cases, a 'staircase effect' becomes visible. It can be assumed, that the surface of the cells are smooth and that the observation of edges or spikes in the segmentation result from imaging artefacts. 2.) The removal of small voxel clusters is especially useful if a threshold close to the background intensity was used. Often single background cluster of the size of a few voxels are assigned to the foreground. These objects do not account for physically present cells in the image. This step reduces also the possibility of one- or two-dimensional objects in the segmentation. These objects can result in inaccurate results and issues in the downstream analysis. Thus, the objects should be removed during the segmentation. 3) The removal of bottom slices. This function emerged due to the observation that objects close to the substrate plane are distorted due to spherical aberrations. This can be explained with the different refractive indices in the glass cover slip and the biofilm-surrounding medium. By removing foreground pixels which are detected in the substrate, effects of this misclassification on the analysis results can be avoided. These tools, however, should be used with caution, because the deletion of segmented objects can introduce biases in the

analysis and should be checked carefully.

The merging of segmentation information describes the processes of pooling the segmentations of multiple channels into a single target channel. It is in particular useful, if no single constitutive signal in a multi-strain – or even multi-species – experiment exists, yet each population has a constitutive signal on its own. If i.e. the local abundance of the individual pseudo-cell populations should be calculated, a signal which comprises all populations is required. The foreground signal of the individual channels are merged into a single pseudo-cell segmentation. In this instance segmentation, a voxel is marked as foreground as soon as it is marked as foreground in any of the involved channels. Given that the involved channels were segmented into pseudo-cells with the same grid size, the object volumes can directly be compared. The local abundance can be derived from the comparison. Depending on the individual segmentation results, the sum of the relative abundances can be larger than one. This is the case if the individual foreground signals overlap. The corresponding pair-wise channel overlap is stored for each resulting pseudo-cell as well. If no pseudo-cells were used, the voxel IDs of all involved objects are merged and the original channel number stored alongside the object properties.

A third-party segmentation, which was imported into *BiofilmQ*, can be inspected by transferring the segmentation channel to the microscope acquisition channel of interest. The voxel IDs and basic segmentation information (i.e. the object volume) of each segmented object is transferred to the desired target channel. Now, the preview function of *BiofilmQ* can be used to inspect the results of the third-party segmentation.

Another post-processing option enables the user to restrict the analysis to a certain number of segmented objects. One application is the investigation of the growth dynamics in smaller biofilms and the interplay between the different physical forces during this initial growth phase [66]. As soon as the number of segmented objects exceeds the user-defined threshold, the analysis is stopped. This limits the costly segmentation to the initial phase of biofilm growth. This can speed up the analysis and reduces the overhead of generating unnecessary segmentation data.

3.6 Parameter Calculation

Although the segmentation of microscope images is and remains a very challenging part in every image processing pipeline, it is only a tool to extract insights into the biological and physical processes in the investigated system. To answer questions of scientific relevance, feature extractions are required. Most feature extraction routines can be reused in different projects. Therefore, new implementations have to integrate with upstream and downstream analysis pipelines (i.e. image segmentation and parameter visualization).

To achieve the demands on the flexibility and to simplify the integration of new features, modularization is required [24]. Thus, the feature extraction in *BiofilmQ*

is organized in independent modules. Each module can be applied on each time frame of a time-lapse experiment independently. The modules are self-containing. The basic features, which are required for all modules, are provided by the segmentation step (Sec. 3.5). If a module requires additional features, the calculation is either implemented in the module itself, or the input is generated by another module which is explicitly called during the execution.

For each time frame and for each channel, the analysis results are stored in a data structure (Fig. 9a). The data structure pools the extracted features with information on the segmentation and acquisition settings for later analysis. Besides the parameters required for the generation of label images (i.e. *Connectivity*, *ImageSize*, *NumObjects*, and *PixelIdxList*), it contains global features of the biofilm which are extracted for the complete ROI (i.e. the biofilm volume, the biofilm surface area, or the mean volume of all segmented objects)(Fig. 9b). The field *MeasurementFields* contains the names of the already calculated features (Fig. 9c). The acquisition metadata (i.e. acquisition data, label for the acquired position, the magnification, or the maximal z-slice number during the acquisition) is stored in the *metadata* structure (Fig. 9d). Since the settings of the image preparation can alter the result of the parameter calculation, the GUI parameters are saved as well (Fig. 9e). The vector list *PixelIdxList* stores the voxel indices for every segmented object. The *stats* table stores the result of the feature extraction of every single segmented object with the corresponding object ID.

The structure is at the same time input and output of each module. Modules only modify the data structure and leave all other GUI parameters untouched. The constant usage of *BiofilmQ* in recent research projects resulted in numerous modules which can be used to extract features from the segmented images (i.e. Tab. 1). The functions of most modules are already covered in the documentation[7]. Therefore, I would like to focus on the modules which are not included in the latest release (v.0.2.1) (marked in Tab. 1). It is to be noted, that the name of a specific module not necessarily overlaps with the name of the extracted parameter. An extensive list of all extractable parameters can be found in Tab. S1 - S4.

Intelligent merging & splitting Without prior fine-tuning, the watershed segmentation creates oversegmented objects [35]. This module tries to mitigate any type of mis-segmentation by using heuristics to identify oversegmented and undersegmented cells [66]. While oversegmented cells can simply be merged with nearby objects, the module uses sophisticated clustering algorithms to split undersegmented cells on a per-pixel basis.

The heuristics are based on the object volume, the object convexity, and the aspect ratios of the height and width of each object. Four rules are in place, each including user-defined threshold values: 1) Objects which have a volume below 0.6 times the median object volume are merged. The users decides whether the object with the largest contact area, or a neighbouring object with the smallest

[7] `https://drescherlab.org/data/biofilmQ/docs/usage/parameter_calculation.html`

Table 1: Parameter calculation modules in *BiofilmQ*. For a full list of the extracted features for each module, see Tabs. S1 - S3. Modules marked with "*" are not part of the latest *BiofilmQ* release (v0.2.1).

Parameter calculation module	**Description**
Default parameter	Parameters directly extracted by segmentation
Filter Objects	Exclude object from analysis based on filtering condition
Remove Objects	Remove marked objects completely from analysis results
Remove Parameter	Remove an already calculated parameter from the analysis results
Intelligent merging/ splitting*	Apply corrections for undersegmented and oversegmented cells
Minimal rotated bounding box*	Calculate corner points of minimal box around object
Size and orientation by ellipsoidal fit*	Calculate lengths and orientation of three rotational axes through object
Aspect ratios	Calculate ratio of rotational axes
Surface properties	Calculate surface properties of biofilm for each grid positions on the substrate
Substrate Area	Calculate biofilm-substrate interface area
Global biofilm properties	Calculate shape related properties of the full biofilm
Alignments*	Calculate object orientation relative to flow direction, z-axis, and radial biofilm direction
Convexity	Object volume divided by volume of convex hull for each object
Distance to centre biofilm	Distance of each object to biofilm centroid coordinate
Distance to nearest neighbor	Distance of each object to the closest object in a user-defined channel
Inter cell spacing*	Distance statistics for object faces
Distance to surface	Distance to outer surface of biofilm
Distance to surface (one side excluded)*	Distance to outer surface of biofilm for large biofilms
Distance to specific object	Distance of each object to a user-defined object
Nemantic order parameter	Alignment of the longest principal axis of each object with the longest principal axis of objects in neighbourhood
Local density	Fraction of occupied space around centroid of each object
Unit cell size	Size of cell in Voronoi diagram
Fluorescent properties	Module which contains all calculations related to fluorescence
Number of fluorescent foci*	Number of fluorescent foci per object
Visualize extra-cellular fluorophores*	Visualization of fluorescence outside segmented objects
Correlation properties	Calculation of correlation measures
Haralick texture features	Texture features
Tag cells	Add user-defined tag to objects based on filtering condition
Custom parameter	Combine parameters with simple mathematical expression to new parameter
User-defined Matlab script	Calculated custom parameter by using a *MatLab* script

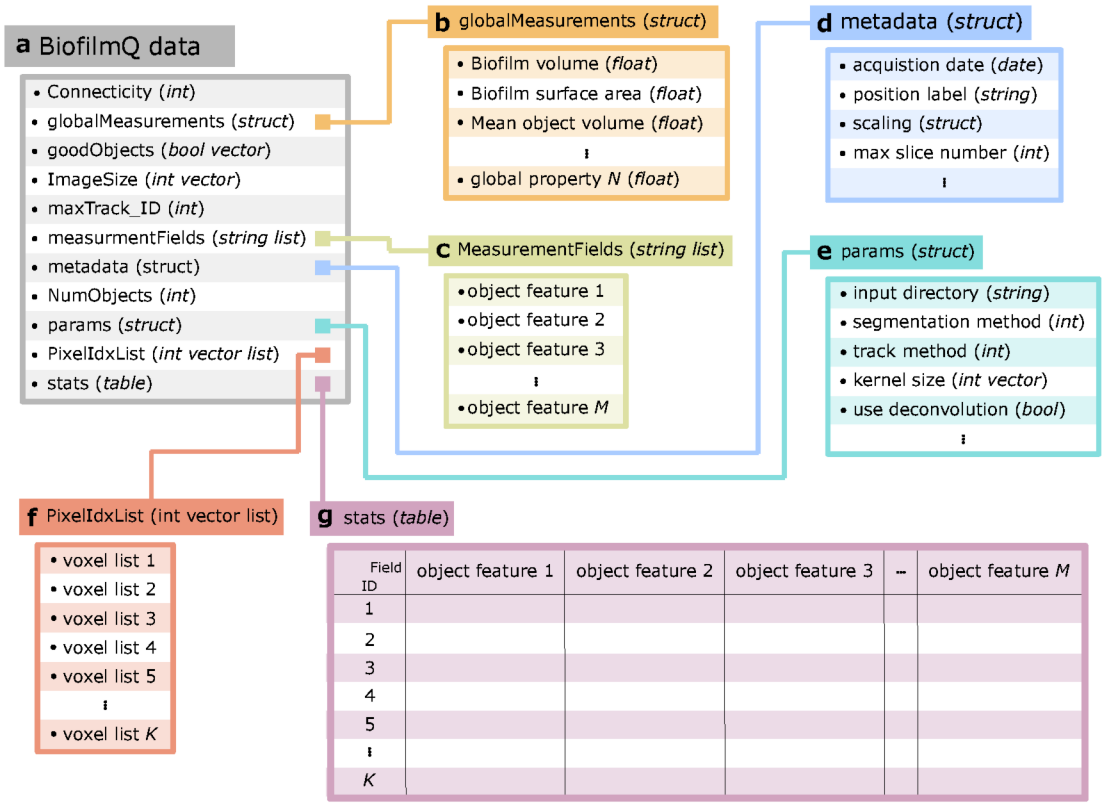


Figure 9: *BiofimQ* data structure. **a** For each time frame and for each channel a separate file is generated. Besides the required information to create a label image (i.e. *Connectivity*, *ImageSize*, *NumObjects*, and *PixelIdxList*), it also contains information on the results of global biofilm measurements (*globalMeasurements*), information whether objects should be excluded from the analysis (*goodObjects*), the maximal lineage ID (*maxTrack_ID*), the names of the extracted object features (*MeasurementsFields*), the acquisition metadata (*metadata*), the GUI parameters (*params*), and full extracted feature list for every segmented object (*stats*). **b** The field *globalMeasurements* contains the full image stack features (i.e the volume or surface area of the imaged biofilm, or the mean object volume for all objects inside the biofilm). **c** *MeasurementFields* contains the field names of all extracted features for quick feature overviews. **d** The metadata of the acquisition are stored in the *metadata* structure (i.e. acquisition date, label for the current position, the information on the scale of a single voxel, or the number of z-slices during the acquisition). **e** The GUI parameters are stored in the *params* structure. This includes all settings for the different *BiofilmQ* modules during the analysis. **f** The *PixelIdxList* contains for each object the index of the member voxels. **g** The results of the parameter calculation are stored in *stats*. For each object ID the corresponding object features are stored.

volume is used as merging partner. 2) If the volume of the object is 2 times larger than the median volume and the object convexity is smaller than 0.7, the object is marked for splitting. 3) Objects with a volume larger than 3 times the median volume are marked for splitting without further conditions on the convexity. 4) Objects with a volume larger than 1.5 times the median volume and with an aspect ratio of the height and width larger than 2 are marked for splitting.

While merging can be performed by assigning an existing object ID twice, the splitting requires a more complex 2-step approach: At the beginning, a k-means clustering algorithm is used to estimate good starting values for a following Gaussian mixture model (GMM) step. For the k-means clustering the number of clusters as well as the feature space has to be prepared.

The number of clusters is initially set to the volume of the merged object divided by the median object volume in the dataset. Additionally, for each object marked for splitting, the core region is estimated by eroding the segmented object with the typical object size. The typical object size is a user-defined property and is equivalent to the edge detection kernel size (compare Sec. 4). If multiple core regions exist for a single object and the number of cores exceed the initially set number of clusters, the number of clusters is updated accordingly.

Besides estimating the core regions of the marked objects, the skeleton is calculated [89] and the angle of each skeleton link with the coordinate system axes (x, y, z) determined. Thus, the feature space of each voxel for the k-means clustering consists of its three spatial coordinates, the three angles of the closest skeleton link to the coordinate axes and the index of the closest object core region.

Finally, a GMM is applied on the plain pixel coordinates. The start indices for each pixel in the GMM search are given by the k-means clustering results. The result of the GMM is directly used to split the object into smaller objects. Since small volumes can emerge due to the GMM splitting, a second merging step is performed on the resulting objects to minimize the occurrence of small objects inside the dataset.

Minimal rotated bounding box This module calculates the dimensions (width, height, and length) of a minimal bounding box fitted around each cell. To calculate the minimal bounding box, the convex hull of the object is calculated first[8]. Since one face of the convex hull has to be parallel with the optimal bounding box base plane, all faces are checked for the minimal volume of the bounding box. The module stores the corner points of bounding box.

Size and orientation by ellipsoidal fit While the position (centroid) of an ellipsoid around the object is directly accessible by calculating the mean position of all voxels inside an object, the orientation and size can be derived from a principal component analysis (PCA). Here, the orientation is given by the matrix of

[8] `https://de.mathworks.com/matlabcentral/fileexchange/18264-minimal-bounding-box`

the Eigen vectors of the covariance matrix. The vectors pointing in the direction of the principal axes and the size (length, width, height) is given by the length of the principal axes of the corresponding ellipsoid[9].

Alignments To measure the orientation of the single cells inside flow chamber experiments relative to the liquid flow, the z-axis, or the radial direction inside the biofilm, an ellipsoidal fit of the object is required. It calculates the relative angle between the longest principal axis of the object and the previously mentioned directions. Here, the flow direction is given by user input and the radial direction is calculated by the direction of the centroid position to the biofilm centre (given by the mean centre position of all segmented objects).

Inter-cell spacing This module calculates the mean value, minimal value, and the standard deviation of the inter-cell distance for each segmented object. The inter-cell spacing is implemented as the length of the normal vector on the centre of a surface patch to the next object. To prevent large computation, the distance is limited by a user-defined search radius. If no other object within the search radius could be found, the object face does not contribute to the inter-cell spacing calculations.

Distance to surface with biofilm on image border Sometimes the captured biofilms are too large in the smallest magnification to fit them into a single field of view of the confocal microscope. To calculate the distance to the surface of such an object, the user selects one border which is not occupied by the biofilm. The holes in the segmented biofilm are filled with a morphological operation and the distance of each object inside the biofilm to the closest non-biofilm voxel is calculated.

Nematic order parameter The nematic order parameter S is defined as

$$S = \left\langle \frac{3}{2}(\hat{n}_i \cdot \hat{n}_j)^2 - \frac{1}{2} \right\rangle, \tag{25}$$

with $\hat{n}_i$ denoting the principal axis of object i. Here, $\hat{n}_j$ are the principal axes of the neighbouring objects within a user-defined range.

Unit cell size This module calculates the volume of a Voronoi cell around the object. If the Voronoi cell could not be created, the data type not a number (NaN) is returned.

[9] `https://github.com/mattools/matGeom`

Extracellular fluorophores Not only the fluorescence of the objects itself, but also the fluorescence around the objects can be extracted. This is in particular useful to visualize the matrix productions of single cells. To use this properly, the matrix component needs to be either stained or the matrix proteins need to be fused to a fluorescent protein. To extract the intensity of produced matrix proteins per cell, a shell of a user-defined thickness is generated around each segmented object. The intensity is extracted and the corresponding information saved in a visualization toolkit (VTK) file which is stored in a separate folder alongside the segmentation information.

Fluorescent foci Besides the mean fluorescent intensity of the cells themselves, *BiofilmQ* supports the counting of fluorescent foci inside the cells i.e. to evaluate the activity of the type-3 secretion system in *Yersinia enterocolitica* [174]. The foci inside an already segmented cell are extracted by identifying local maxima. The maxima are spotted by dilating the channel of interest with a structural element of a user-defined size and using all values which are above the original intensity. The contrast between the maxima and the surrounding pixels is calculated and an Otsu threshold [128] applied to filter out FP detections.

3.7 Object Tracking

Single-cell tracking in dense biofilm communities adds additional complexity to the image analysis pipeline. However, it enables quantitative assessment of growth rates and velocities of single cells which are not accessible otherwise [66], i.e. influence of second messenger molecules such as cyclic diguanylate (cyclic di-GMP) on the motility of cells [182], investigating the influence of environmental queues on chemotaxis [126], or analysis of cell fate after multiple cell division events [100]. Simple algorithms use the overlap of objects in different time frames to reconstruct the trajectory of single objects. Early attempts have been made to maximize probability distributions [36], yet the computational expenses were too large to use this approach on entire volumes. This most likely constitutes the reason why previous attempts limited themselves to early developmental stages of biofilms, or used fluorescently labelled subpopulations [148] or else, fluorescent foci [134]. These approaches do not solve the tracking for all objects inside the imaged volume, but rather focus on single tracks. In general, the migration patterns of selected cells are good representations of the overall cell migration inside the specimen. Only recently, an overlap-based algorithm attempted to assign cells to a single founder cell inside 3D biofilms [66]. However, the approaches lacked the sophisticated algorithms used to track cells in eukaryotic systems which have been used to assign cell fate maps for a large number of cells inside mouse embryos [113, 6].

While the tracking of eukaryotic cells made extensive progress in recent years [160], the tracking of single cells inside biofilms remains challenging. There are three major difficulties in imaging dense microbial communities: 1) The needed resolution for imaging single-cell objects inside biofilms to the optical limit of

the used equipment[132]. 2) The dissection of objects in low contrast images is difficult. 3) The lack of publicly available test sample datasets which prevented algorithm optimizations.

Two different tracking approaches are implemented in *BiofilmQ*. One algorithm focuses on the tracking of segmented single-cell objects in time-series experiments, the other algorithm performs tracking of pseudo-cell positions. Regardless of the underlying object type, both algorithms rely on existing tracking information in previous time frames. If the frame is the first frame in a time series, new tracking IDs are assigned.

Single-cell tracking A step-by-step explanation of the different cases and assignment rules implemented in the cell tracking algorithm is shown in Fig. 10. For illustrative purposes the 3D approach is presented in a 2D example. Initially, only the instance segmentations in time frame t are available (Fig. 10a). Together with the already existing lineage information in the time frames $t-1$ and $t-2$ (Fig. 10b) the lineage information in frame t can be restored. To increase the possibility of an overlap between the objects in different time frames, all objects are dilated by a user-defined number of pixels (not shown).

In a first step the objects in time frame t are compared with objects in time frame $t-1$ (Fig. 10c). For each cell in frame t, all cells in frame $t-1$ whose centroids are within a user-defined distance (left) are considered as predecessor candidates. The overlap o_i of the object with the candidate objects i are calculated. Additionally, the relative angle between the longest principal axis α_i of the candidate and the investigated cell is extracted (right). Based on these two parameters, a simple cost function

$$c_i = o_i + o_i \cdot \frac{\cos(\alpha_i)}{2} \tag{26}$$

is defined. Here, o_i denotes the overlap of the current object with the dilated object i and α_i is the angle difference between the longest principal axis of both objects. The candidate with the smallest cost is considered as predecessor and its track ID assigned to the current object.

For all objects in t that could not be assigned to a track ID, the algorithm rescans the current time frame t whether neighbouring objects exist (Fig. 10d). Neighbouring objects are identified by overlapping pixels. If the neighbouring objects already have a track ID, a split of the parent object can be assumed and the track ID is assigned to both objects (left). If only neighbouring objects without track ID are found, the objects get assigned to a new track ID (right).

For the objects which still do not have a track ID, the algorithm searches for objects in frame $t-2$ for possible pre-predecessors (Fig. 10e). Again, all objects whose centroids are within the search radius around the centroid position of the untracked object in frame t are considered as possible candidates (left). Similar to Fig. 10b, the overlap and the angle between the longest principal axes are calculated. To prevent FP detections however, only candidate objects with an

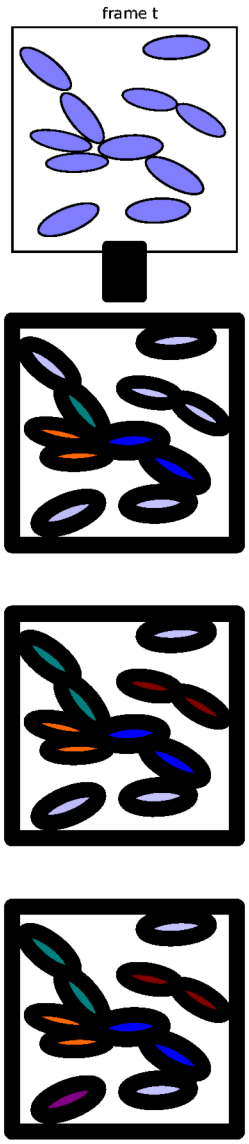


Figure 10: Overview of cell tracking algorithm (here, 2D). **a** Each object in the current time frame t is assigned to a tracking ID in four steps. **b** Tracking data of the previous time $t-1$ and $t-2$ is available and used for tracking (indicated by colours). **b** For each object, predecessor candidate centroids within a user-defined radius are searched. Candidates are selected such that the overlap is maximized and the orientation difference minimized. **d** Algorithm checks for neighbouring objects. Left, if a neighbour with a track ID exists, the same track ID is assigned. Right, if the neighbour does not have a track ID, a new track ID is assigned to both objects. Objects without neighbours remain unchanged. **e** For objects without track ID, the time frame $t-2$ is used. Right, cells are only matched if orientation is almost identical and overlap is more than 50%. **f** If no predecessor could be found, a new track ID is assigned.

almost parallel axis

$$\cos(\alpha_i) > 0.8 \tag{27}$$

and an overlap of at least 50% are considered (right). This tracking step prevents lost lineage information due to FN detections by the used segmentation algorithm. If no predecessor or pre-predecessor object could be found, the object is considered as a new object (i.e. a cell which attached at the position between the time frame t and $t-1$) and gets a new track ID (Fig. 10f).

Although the presented algorithm is quite simple, its development is based on heuristics extracted from numerous biofilm experiments. It reliably reconstructs lineages for small biofilms and can successfully discriminate against cells which did not originate from a single founder cell in long time-lapse experiments [66]. To identify those invading cells, it is not necessary to track every single cell inside the bulk biofilm itself, but rather track only a cluster of cells over time. To support this quantification option, the user has the option to start a new track at any time frame and assign every cell in the starting frame the same track ID. Thereby, invading cells which attach to the biofilm at a later time frame get assigned to a different track ID and can easily be identified and excluded from the biofilm growth rate measurements.

Pseudo-cell tracking If the experimental setup is not suitable for single-cell resolution or only the extraction of the growth rate of the entire clonal population is required, the occupancy of single positions inside the 3D structure can be used as a proxy for lineage tracking. For this purpose, *BiofilmQ* contains a pseudo-cell tracking algorithm. The implemented steps are described in Fig. 11. Similar to the illustration for the single-cell tracking, a 2D example for the 3D tracking is shown.

Initially, only a foreground-background segmentation for time frame t and the corresponding dissection into pseudo-cells is available (Fig. 11a). By using the available tracking information in the previous time frames $t-1$ and $t-2$, track IDs can be assigned to all foreground objects in time frame t. The approach consists of three stages.

In the first stage, the current time frame t is compared with the previous time frame $t-1$ (Fig. 11b). Grid positions which had a tracking ID in frame $t-1$ are assigned to the same ID. Since the pseudo-cell dissection is cell-agnostic, some pseudo-cells share the same track ID, although the corresponding position contains voxels of two different physical objects (i.e. *e4*)). In contrary, pseudo-cells which belong to the same physical object are not assigned to the same ID (i.e. cell with centroid in *i6*). This approach is repeated with time frame $t-2$ (Fig. 11c).

In the next stage, the information about the neighbouring volumes of each unassigned pseudo-cell is used. If only interfaces to objects with the same track ID exists, the track ID is used for the object as well (i.e. *6c* in Fig. 11d).

If multiple possible track IDs exist, (i.e. *red*, *green* for position *d6* in Fig. 11e) the

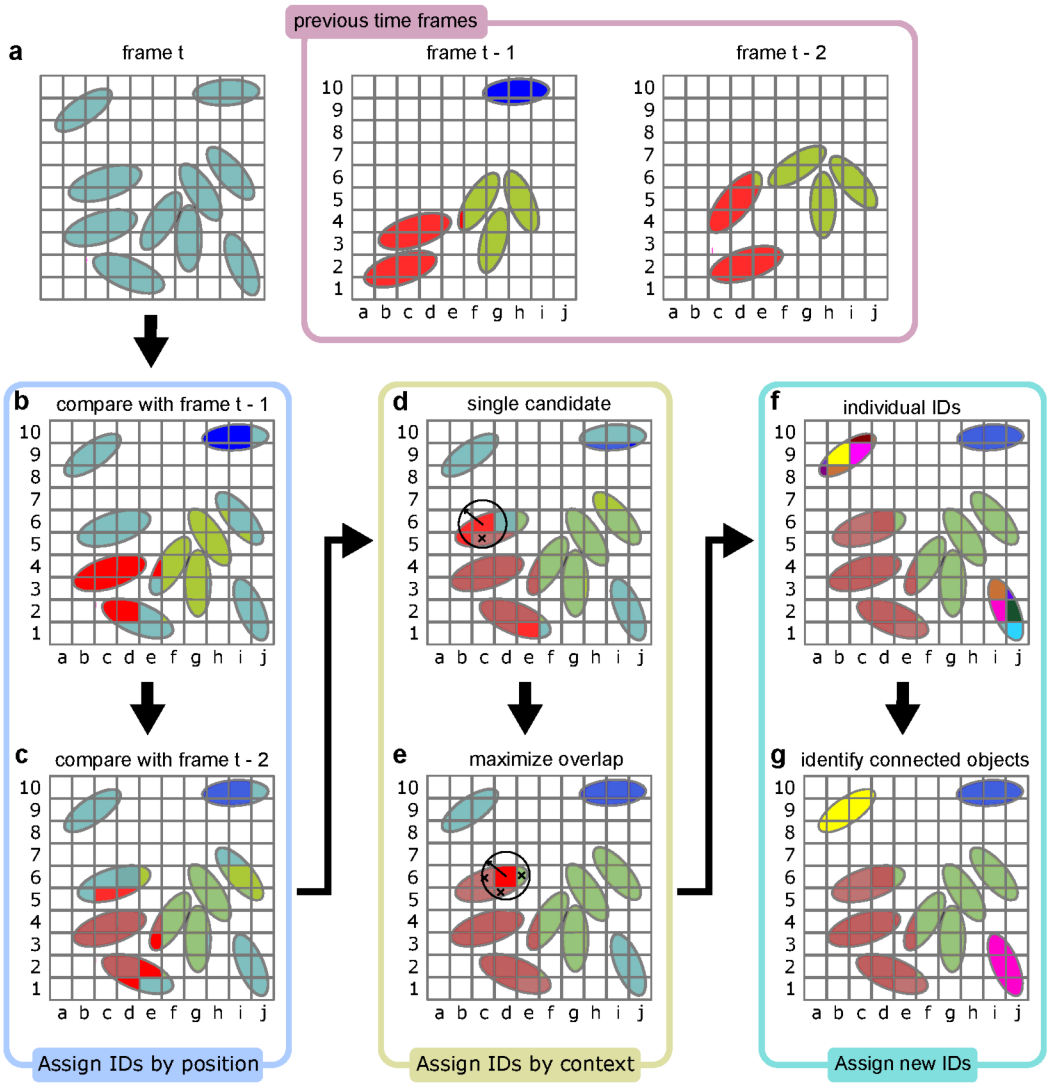


Figure 11: Overview of the pseudo-cell tracking algorithm. **a** New time frame t is analysed, the track IDs for the previous time frame $t-1$ and $t-2$ already exist. **b** In a first attempt, the occupied positions in $t-1$ are compared with the positions in t. **c** Positions without track ID are compared with occupied positions in frame $t-2$. **d** For position which do not have a track ID yet, the centroids within a search radius are sampled for a track ID. **e** If more than one track ID exists, the track ID of the object with the maximum overlap with the investigated volume is used. **f** All still unassigned volumes receive new track ID. **g** Pseudo-cells in proximity are assigned to the same track ID.

foreground fraction of the position and the foreground fraction of its neighbours are dilated by a user-defined number of voxels and the algorithm calculates the overlap between the dilated objects. The track ID with the largest number of overlapping voxels is assigned to the pseudo-cell.

If none of the described steps leads to a tracking ID for a detected pseudo-cell, a new ID is assigned (Fig. 11f). As an additional feature the user can enable the identification of connected objects based on a user-defined search radius (i.e. Fig. 11g). This feature enables the identification of cell clusters and assigns all objects inside the cluster to the same new tracking ID. With the presented approach, the growth rate of clonal populations can be calculated for different time steps even if they merge to an indistinguishable biomass cluster.

3.8 Data Export

During the analysis, the segmentation and object data is stored in the *Matlab*-specific MAT file format in a separate folder. In order to access the results outside the presented GUI, *BiofilmQ* provides export functions to standard file formats. The results of the parameter processing can be exported to a comma separated values (CSV) file, which is accessible i.e. with a simple text editor and other data analysis programs or libraries.

For the exchange of cytometry data the flow cytometry standard (FCS) was established [153, 122]. Here, every extracted object parameter is comparable to a single channel of a flow cytometry experiment and thus can be stored in the corresponding exchange format.

3D renderings provide a powerful tool to visualize the extracted features of a biofilm in 3D. One 3D data exchange format is VTK [144]. By exporting the segmentation data, each object is stored with the measured properties and can be visualized in the 3D rendering software *ParaView* [4].

3.9 Plotting

One main purpose of the *BiofilmQ* project was a shorter time between image acquisition and the first analysis results i.e. visualizations. This is in particular important to quickly pivot or adapt the experimental settings if necessary and to get a quick overview of the collected data. Therefore, *BiofilmQ* provides a large variety of plotting options to visualize the extracted features (Fig. 12).

A condensed view of the visualization tab is shown in Fig. 12. It contains different input fields to select parameters, set plotting options or plotting axis, and to select a subset of the extracted data by gating.

The user has the option to switch between the single object properties and full biofilm parameters (not shown). To shorten the search time for a certain parameter, the search field can be used. A selected parameter can be applied quickly

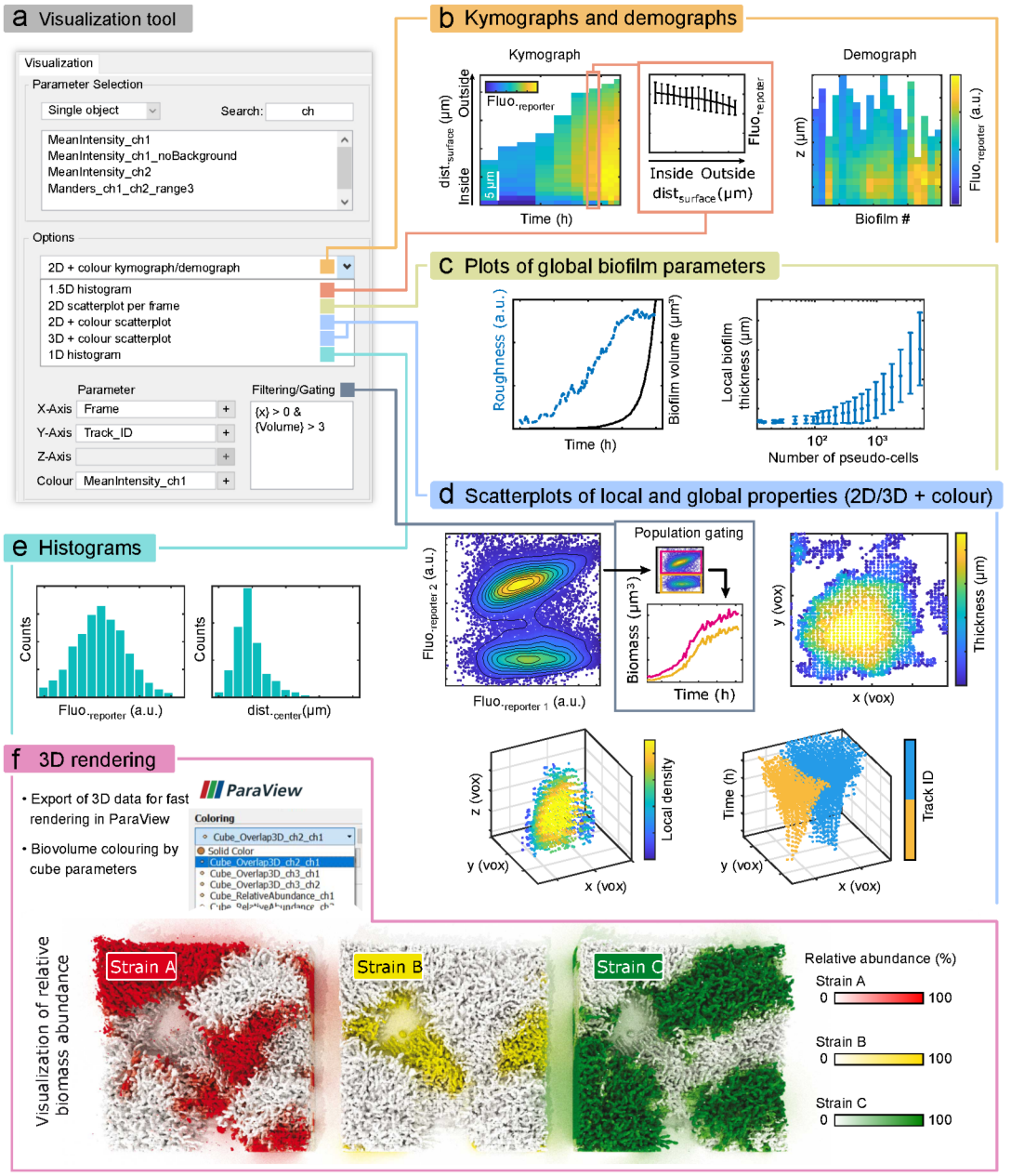


Figure 12: Overview of the available plot types in *BiofilmQ*. **a** Panel for plotting settings. **b** Left, heatmap shows spatio-temporal activity of a fluorescent reporter (here: RpoS translational fusion). Centre, a histogram of the fluorescent reporter activity over the spatial component at a selected time. Right, Comparison of fluorescent reporter activity in biofilm replicates. **c** Left, comparison of biofilm roughness and volume over time. Right, local biofilm thickness over number of objects. **d**, Top left, image cytometry for two different fluorescent channels. Top centre, gating splits two populations and visualize development of biomass (by measured volume) separately for each gate. Top right, 2D plot shows biofilm thickness for each substrate positions. Bottom left, 3D colour plot shows spatial density distribution in 3D. Bottom right, visualization of merging cell clusters determined by position-based tracking. **e** Visualization of fluorescent reporter activity and distance to centre for a single biofilm. **f** *ParaView* visualization of spatial abundance of three different strains in 3D. Figure adapted from [65].

to one of the available plotting axes. In the option panel, the user can select one plotting type for the visualization. Depending on the selected type, the different axes can be filled with the extracted parameter. The user can use the symbol next to the text field to assign the currently selected parameter. Without user-defined gating, all objects are used in the visualization. The user can apply the gating options by using the *Filtering/ Gating* text field on the left.

The plotting examples are presented in Fig. 12 panels b-e. A well-established practise to present spatio-temporal activity of fluorescent reporters are heatmaps [41, 179]. The heatmaps collapse the three spatial dimensions to a single dimension based on a distance measure (i.e. Fig. 12b right). Here, every bar represents a histogram for the parameter on the colour axis with respect to the parameter selected for the vertical axis (Fig. 12b left). This can also be used to create demographs, i.e. comparing different biofilms in different positions, different channels, or even different experimental settings with each other (left).

Very useful to observe dynamics of changing parameters over time are 2D plots. They accumulate individual object related parameter measurements to a single data point at a given time (Fig. 12c). Additional features include the option to accumulated multiple plots in a single canvas (left) or to use the number of segmented objects as a (quasi-)time.

Scatter plots can be used in the 2D and 3D variant with an additional colour dimension (Fig. 12d). The examples for such plots range from plots similar to those typically used in flow cytometry experiments where two fluorescent intensities are assigned to the coordinate axes (left). Here, i.e. gating can be applied to extract not only population abundances in certain regions but extract any individual measured parameter into a separate plot (centre). Scatter plots can also be used to visualize extracted parameters in 2D and 3D (top right, bottom left) or even provide an overview of two merging cell clusters over time (bottom right).

An alternative to the common histograms (Fig. 12e), is the visualization of the measured single-cell parameter in a 3D rendering in *ParaView* (Fig. 12f)."

3.10 Batch Processing

Modern microscopes are fully automatized. Equipped with motorized stages, auto-focusing systems, incubator chambers, and connected to large storage devices, the gathered datasets quickly accumulate to hundreds of gigabytes. Enabled by the advances in imaging techniques, the user can easily collect 20-50 technical replicates per experiment. While a single replicate can be analysed by hand, the manual analysis of entire transposon mutant library scans distributed in several hundred 96 well plates is almost infeasible.

Therefore, *BiofilmQ* comes with three different levels of batch processing capabilities (Fig. 13). The fully manual, step-wise case was presented earlier in this section (Fig. 13a). The user interacts with the GUI, i.e. manually sets parameters

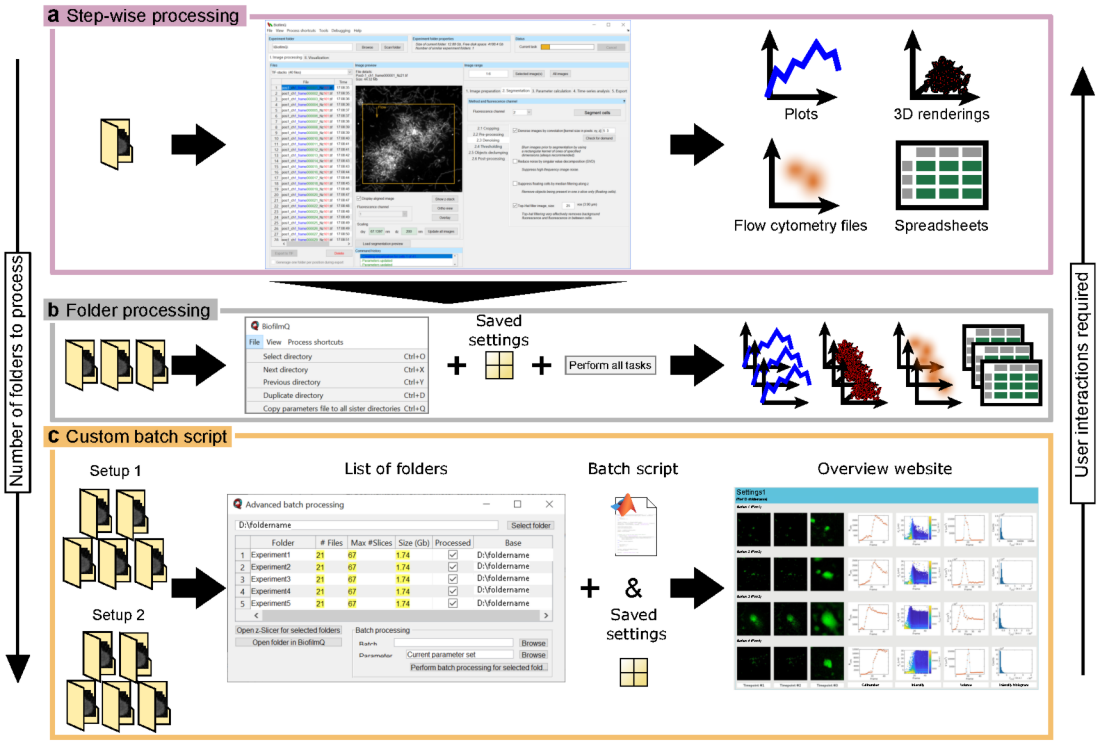


Figure 13: Overview of *BiofilmQ* batch processing. **a** The simplest workflow for processing image data in *BiofilmQ* relies on step-by-step GUI interaction. This is the preferred approach for analysing a single experiment folder or to test different settings for more advanced batch processing steps. Basic GUI interactions generate a limited number of analysis results in the shape of plots, 3D renderings, FCS files, and CSV files which are all stored alongside the image data. **b** A more advanced batch processing of multiple folders within the same parent directory can be used to replicate analysis steps for multiple technical replicates. The parameter settings can be generated by using the GUI and re-used for multiple folders. This enables the user to quickly analyse a relatively large number of folders with little to none user interaction within the GUI. The user navigates to the correct experiment folder and starts the processing with a single batch processing button. The analysis results are saved in the individual experiment folders. **c** For a large number of experiments with similar but not identical analysis settings, a custom batch script can automate the data analysis. This is the preferred approach for analysing multiple folders from different biological replicates with several technical replicates each. An additional batch processing dialogue assists the user in the set up of the workflow. Besides a parameter file for the default parameters, a custom batch script is needed. In the batch script, the user can implement custom processing routines, modify the parameter setting for different experiment folders based on external inputs, or just define the sequence of *BiofilmQ* analysis routines which should be applied. *BiofilmQ* includes batch files for the most common applications. Since this approach can quickly generate large amounts of analysis results, *BiofilmQ* can generate an overview website which gathers all generated visualizations in one place.

for the different analysis steps and starts them one after the other by pressing the corresponding function buttons. This is only advisable for the set up of the image pipelines for small datasets. Otherwise, the waiting time between the processing steps diminishes the processing advantages and no additional productivity gain can be expected.

Once the parameters are set and the analysis pipeline is working for one example dataset, the user can reuse the saved GUI settings and apply them to other folders within the same parent directory (Fig. 13b). By selecting the desired tasks, the software provides a trigger to run them without any further interactions.

While this already results in a significant analysis speed-up, this option does not provide the flexibility which is required for large-scale experiments: First, it does not offer any customization option for different folders. Second, it cannot process entire folder structures at once; the user still has to change the folders manually.

To solve this, *BiofilmQ* provides a folder overview dialogue which lists all experimental folders (i.e. folders which contain files which correspond to the expected naming scheme as described in Sec. 3.3) under a selected root folder (Fig. 13c). The user can select the folders which shall be processed by the program, define a default parameter file which is applied to all experiment folders, and implement a fully customizable batch script to automatize the analysis. Since the creation of such a batch script requires in-depth knowledge of the *BiofilmQ* architecture, batch scripts for the default analysis steps are already provided. Guided by these templates, *BiofilmQ* can analyse entire folder structures with different experiments in one go. During the process, no user interaction is required. The process can easily be scaled from a single desktop machine to an entire compute cluster.

Since the analysis can potentially create large amounts of data, which are scattered in different analysis subfolders, *BiofilmQ* includes a batch script which generates an overview website in hypertext markup language (HTML) format. The website contains all analysis results on a single page and is accessible with a standard web browser. With this feature, outliers can be quickly identified and large biofilm experiments reviewed.

3.11 Conclusion

BiofilmQ is a powerful tool for the analysis of fluorescent microscope images. With its rich set of feature extractions and image segmentation techniques, it enables researchers to quickly import, analyse, visualize, and assess the results of fluorescent microscopy experiments. In contrast to existing analysis libraries, *BiofilmQ* provides the full analysis capabilities in a single GUI. Thus, the full spectrum of features is accessible without writing a single line of analysis code and can generate publication-ready figures.

The pseudo-cell segmentation approach enables the spatio-temporal analysis of dense biofilm communities which are inaccessible for single-cell analysis. The comparison of pseudo-cell image cytometry and flow cytometry for macroscopic

biofilms indicated that both methods can obtain qualitatively similar results. When compared with image cytometry based on single-cell segmentation, the image cytometry on pseudo-cells produced closely matching results. In contrast to flow cytometry approaches, image cytometry investigations offer the additional benefit of preserved spatial information and does not require the disruption of the specimen prior to the analysis. This enables spatio-temporal investigations of fluorescent patterns inside dense microbial communities which are inaccessible using single-cell methods.

Due to the modular feature extraction in *BiofilmQ*, new analysis ideas can be quickly implemented and benefit from the already available functions for file import, visualization, and large-scale batch processing. With the cell tracking features, migration patterns of single cells or cell clusters can be investigated over time. Based on these tracks, growth rates can be calculated and the reconstruction of lineage trees inside biofilms come into reach.

With the application of the described features, *BiofilmQ* has already proven its usefulness in recent publications on the field of bacterial biofilms [38, 150, 129, 59, 92]. Based on the experience in an inter-disciplinary research group, the provided features are a valuable extension to the image analysis landscape and I expect that *BiofilmQ* will be useful to many other research groups on the field of bacterial biofilms.

4 Single-Cell Segmentation

4.1 Introduction

The foundation of today's image analysis algorithms for bacterial biofilms is based on studies on spherical particles in colloidal suspensions [36]. In this study, the position and trajectory of spherical objects were extracted from digitalized microscope recordings. Although limited in the technical equipment and available compute resources, the precise presentation of the used algorithms enabled researchers to quickly adapt the used image analysis techniques. After the extension to rod-shaped particles and the introduction of CLSMs, the framework was not only capable to determine position and trajectory, but also the orientation of single objects in proximity to each other [119]. With these tools, first investigations of single cells inside bacterial biofilms extracted individual alignment and inter-cell spacing, and opened the door for single-cell bacterial biofilm research [43, 154]. However, the investigations focused exclusively on orientation, position, and number density in a small part of a full biofilm.

In order to investigate physical interactions, the wall-to-wall spacing as well as the cell shape are important parameters. However, the previously mentioned investigations could not capture the volume of the single objects on a per-voxel basis. This changed by using automatic threshold algorithms for the segmentation of individual z-planes in stained biofilms [41]. By stitching the individually-segmented slices together, complete *Vibrio cholerae* biofilms could be investigated in all spatial dimensions [41]. For the first time, different architectural phases during the biofilm development could be observed and attributed to the behaviour of single cells. Due to the used deoxyribonucleic acid (DNA)-binding dye (SYTO 9), however, the investigation of the temporal development was limited to ensemble averages. Thus, a large quantity of experiments was required to extract meaningful insights into the different developmental phases. This changed with the successful imaging of fluorescent reporter strains [179]. Finally, the imaging of living biofilms in space and time was possible. This was achieved by building on previous modifications in the used microscope hardware [41] and by using deconvolution algorithms for enhancing the image quality prior to the segmentation [179]. The problem of under-segmentation was solved by using the boundary information of a watershed transformation in every z-dependent threshold step[10].

Building on the techniques developed in previous publications [41, 179], edge detection algorithms replaced the simple z-plane-dependent threshold segmentations. Additionally, the undersegmentation problem was mitigated by using k-means clustering to separate larger cell clumps into individual cells. This enabled the investigation of phage protection mechanisms by curli-fibers and the investigation of stress response under glucose starvation in *Vibrio cholerae* biofilms on a single-cell level [164, 151].

Currently, edge detection-based algorithms produce single cell detections inside

[10] `https://github.com/yanjing32/Single-Cell-Tracking`

bacterial biofilms [66, 38]. These algorithms achieve appealing results by leveraging heavy post-processing procedures to split under-segmented cells with k-means clustering, GMM, and topological skeletonization. The authors tested the accuracy on synthetic data and estimated an accuracy of more than 95% by using simulated biofilms with artificially introduced noise and PSF-based blur. The accuracy estimation was based on the direct comparison of the reconstructed and simulated dataset in cell number, displacement to the nearest corresponding cell neighbour, and orientation of the mean axis [66]. While those parameters were most relevant for the published findings, a rigorous accuracy estimation based on the commonly used F_1 score (20) or average precision is not available.

Other research groups successfully created single-cell segmentations inside bacterial biofilms [168, 183] by using an *U-Net*-based semantic segmentation [48] and graph cuts to mitigate the emerging under-segmentation. They estimated an average precision at an IoU of 0.5 as between 0.35 - 0.6 for experimental biofilm data, while achieving almost perfect scores on simulation data. The simulated volumes only consisted of hundreds of cells, while commonly observed biofilms inside flow chambers contain several thousands. In these imaging heights the available algorithm by Hartmann et al. also exhibit strong segmentation performances. However, the exact average percision of Hartmann et al. on experimental remains unknown.

Although each add-on to the existing algorithms pushed forward the discovery of new biological or physical interactions inside biofilms, sophisticated cell segmentation can be circumvented by using biological markers. The markers simplify the tracking of single cells in a dense population. Cell lineages can be tracked i.e. by a constitutive mNeonGreen-μNS (mNG-μNS) fluorescent protein, which forms a punctum in each cell [134]. Yet, the single points do not capture orientation, or shape related properties of the individual bacteria and are limited to a single daughter cell after mitosis. This limits the number of extractable parameters and gives rise to a large number of cells inside the biofilm with unknown lineage information.

This chapter provides a rigorous average precision measurement (19) of the single-cell segmentation algorithm by Hartmann et al. [66]. To evaluate possible improvements in the segmentation pipeline, I discuss the implemented processing steps. The chapter starts with the analysis of the edge-detection-based semantic segmentation (Sec. 4.2). The semantic segmentation is followed by a watershed-based instance segmentation which relies on a custom watershed implementation (Sec. 4.3). The accuracy evaluation is performed on the direct segmentation output and the in Sec. 3.6 discussed post-processing approach (Sec. 4.4). Finally, I propose an extension to the algorithm and show that superior segmentation accuracies with respect to F_1 score and average precision can be achieved (Sec. 4.5). This enables the path for more accurate measurement of spatial properties of single cells inside bacterial biofilms.

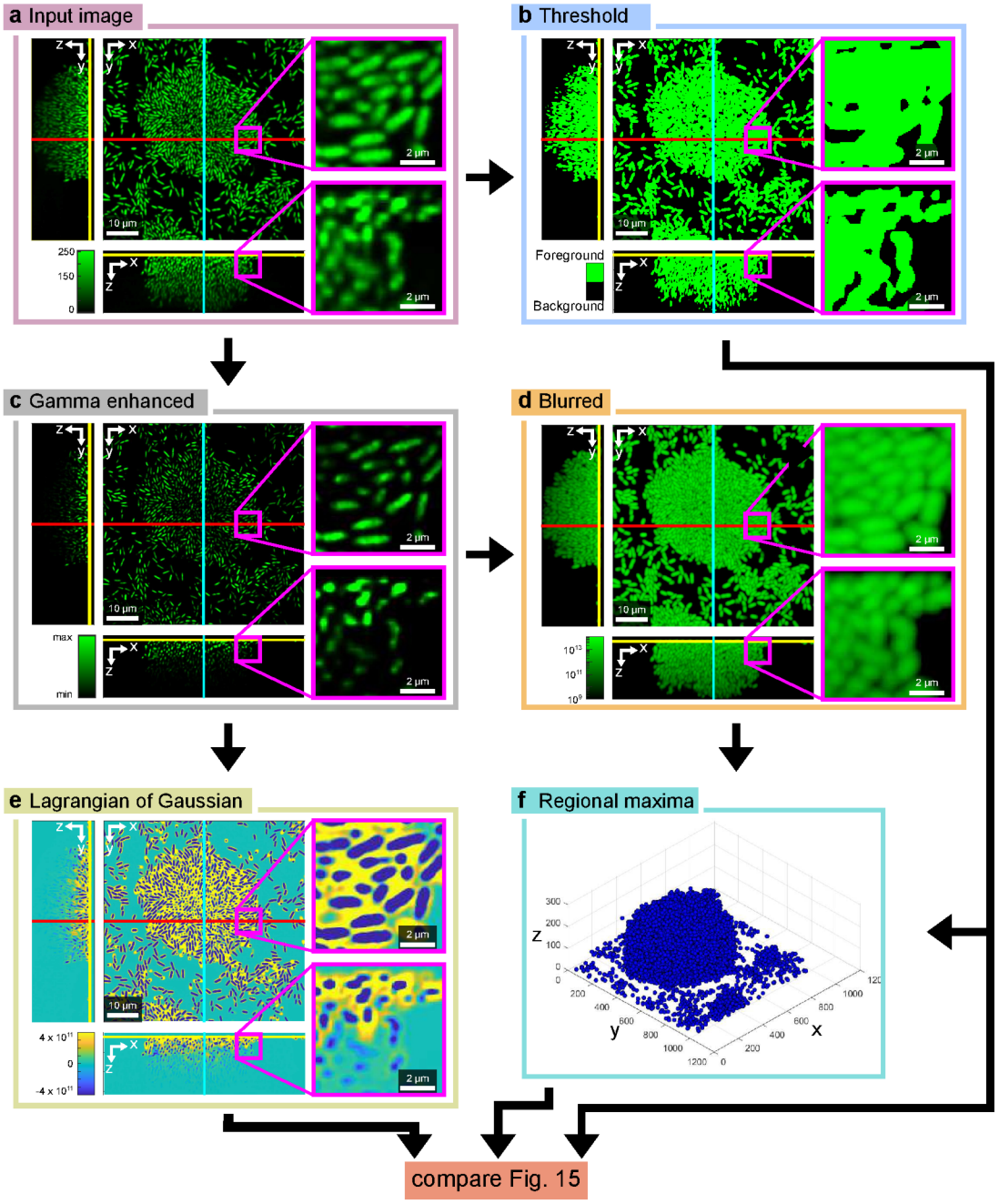


Figure 14: Edge detection approach for semantic segmentation (Part 1). **a** Deconvolved and noise-reduced image serves as input data. **b** A simple Otsu threshold is used to exclude background voxels from further downstream analysis and provides a rough estimate of the foreground voxels. **c** Setting the intensity information to the power of a user-defined exponent enhances the edges in the image stack and enlarges the inter-cell distance for single-cell segmentation. **d** To extract the cell centre positions, the image is blurred with a Gaussian kernel to reduce the intensity fluctuations in the captured fluorophore emissions. **e** With a LoG kernel the single cells are assigned to negative values, while the neighbourhood of every single cell is assigned to positive voxel values. **f** Based on the blurred image, regional maxima are extracted. Local maxima in background voxels are omitted. The semantic segmentation pipeline proceeds in Fig. 15.

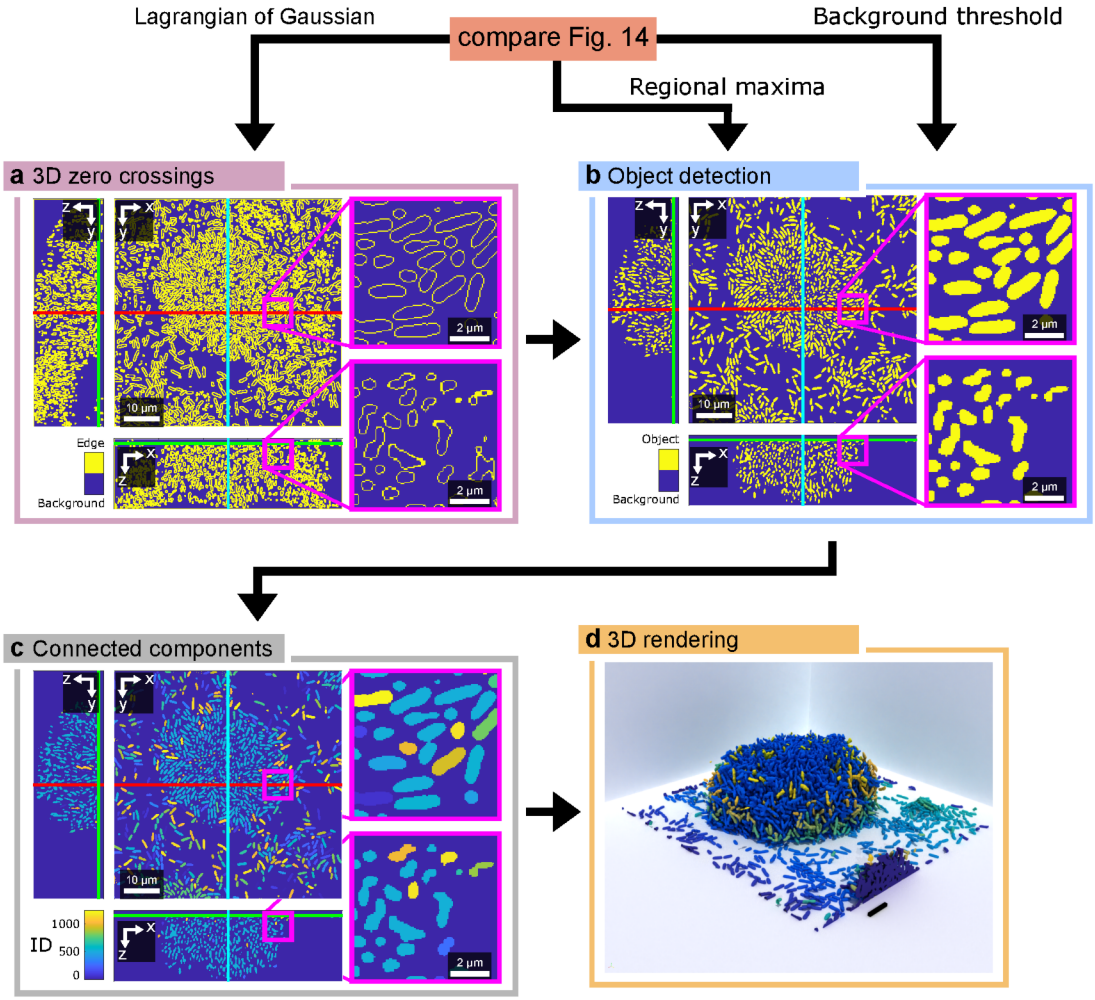


Figure 15: Continuation of semantic segmentation in Fig. 14. **a** In the result of the LoG filter, cell edges are given by iso-surfaces with the value zero. **b** The detected edges are restricted to edges in foreground voxels and the previously calculated regional maxima are used for a seed-based flood filling of the cell shapes. **c** Although the cell shapes are well separated, large clusters exist in 3D and further instance segmentation is needed to mitigate the over-segmentation. **d** The extracted cell shapes already give an excellent representation of the biofilm in 3D (scale bar equals 5 µm).

4.2 Semantic Segmentation with Edge Detection

The algorithm employed by Hartmann et al. builds on segmentation pipelines used in previous publications [164, 151], and first implementations can be tracked back to the edge detections for single-cell segmentations in the PhD thesis of the same author [64].

To perform the segmentation steps, a denoised image is needed. The exact reconstruction of the physical cell shapes benefit from a previous deconvolution step (Sec. 2.5). The basic workflow is depicted in Fig. 14 and Fig. 15 and consists of a self-contained pipeline to generate an accurate semantic segmentation. Starting from the deconvolved and noise-reduced input volume (Fig. 14a), a global threshold is calculated based on a three-class Otsu algorithm. Here, everything below the lowest threshold value (i.e. the first class) is considered background, while everything higher than the threshold is used as a rough foreground estimation. During this processing step, the accurate detection of background voxels is emphasized. Voxels which are marked as background voxels are ignored in the downstream analysis.

To enhance edges in the volume, the individual pixel intensities are raised to the power of a user-defined exponent (Fig. 14c). The higher the exponent, the more pronounced the edges become, but the thinner the cells appear. This can result in cell shapes which do not represent the physical objects any more. Thus, large exponents should be avoided, if an exact volume extraction for single cells is needed.

Based on the pronounced edges, a Gaussian filter with a sigma value close to the typical cell width is applied to the dataset. This blurs the cells and pronounces the cell body in comparison to the edges. Additionally, the number of local maxima, which emerge due to the fluctuations in the fluorophore emissions, is reduced (Fig. 14c).

The blurred volume is used to extract local maxima inside the stack. This method is currently very sensitive to small intensity changes. Therefore, local maxima can arise inside the cell body as well as in background voxels. Thus, the rough background segmentation (Fig. 14b) is used to clear local maxima in known background voxels. Only the maxima in foreground voxels remain (Fig. 14f). However, multiple maxima inside a single cell can persist. These inhomogeneities in the captured fluorophore brightness can be explained by statistical effects in the photon emission of the fluorophores – especially for low SNRs.

At the same time, a LoG filter with a kernel size similar to the typical – yet user-defined – cell size is applied to the edge-enhanced dataset (Fig. 14e). In the resulting image stack, voxels which are associated with the cell cytoplasm get assigned to negative values, while darker background pixels are assigned to positive values. The results of the LoG filtering (Fig. 14e), the local maxima estimations (Fig. 14f), and the result of the global threshold calculation (Fig. 14e) are used for further downstream analysis steps which are shown in Fig. 15.

Based on the LoG results, cell outlines can be extracted. Since the cells are marked with negative voxel values and voxels outside the cells are assigned to positive values, the cell outlines are located on iso-surfaces with the value zero. The iso-surface can be obtained by searching for the sign flips between neighbouring voxels. If the sign differs, an iso-surface with value zero exists between the investigated voxels. A special case is a voxel with a zero value. For a true zero crossing, a sign flip between the voxels on opposite sides of the investigated voxel have to be present. As shown in Fig. 15a zero crossings not only exists around the single cells, but also in locations outside the biofilm.

To filter edges which are not associated to a fluorescent cell signal, the previously captured threshold information is used (Fig. 15b). With the additional information on the regional maxima, a seed-based flood filling of the iso-surfaces can be performed and gives rise to robust foreground voxel detection. In contrast to a threshold approach, the identification of the cell outline with edge detection gives much cleaner results on the cell shape and is not effected by inhomogeneous background intensities.

Although the extracted foreground voxels resemble single cells (especially in the xy-plane), the identification of connected components inside the dataset reveals large under-segmented cell clusters (Fig. 15c). Due to small bridges between the cells, large clusters span the complete size of the biofilm. This becomes apparent when the semantic segmentation result is rendered in 3D (Fig. 15d). The separation between the clusters can be enhanced by a larger exponent in Fig. 14c. With a large exponent, the connected component analysis can identify individual – yet distorted – objects more easily. Even with an intermediate exponent, the semantic segmentation produces high-quality results. Due to the under-segmentation, however, further processing steps are needed to extract volumetric single-cell information from the dataset.

4.3 Watershed-based Instance Segmentation

As explained in the previous section (Sec. 4.2), the edge-detection-based semantic segmentation by Hartmann et al. is – without further instances segmentation – unusable for the extraction of single-cell features due to severe under-segmentations. To mitigate the under-segmentations, a watershed segmentation is used (Fig. 16).

Besides the already deconvolved and noise-reduced intensity volume, the under-segmented result of the semantic segmentation (Fig. 16a left) is needed to build the watershed topology map. The watershed segmentation uses the differences in the intensity levels in the fluorescence channel to split large under-segmented objects (i.e. yellow object in Fig. 16a right).

From the previous semantic segmentation step, the regional maxima (Fig. 16b) can be reused to deepen the „valleys“ of the watershed topology. The watershed energy landscape (or topology map) is created by enhancing the pixel intensity at

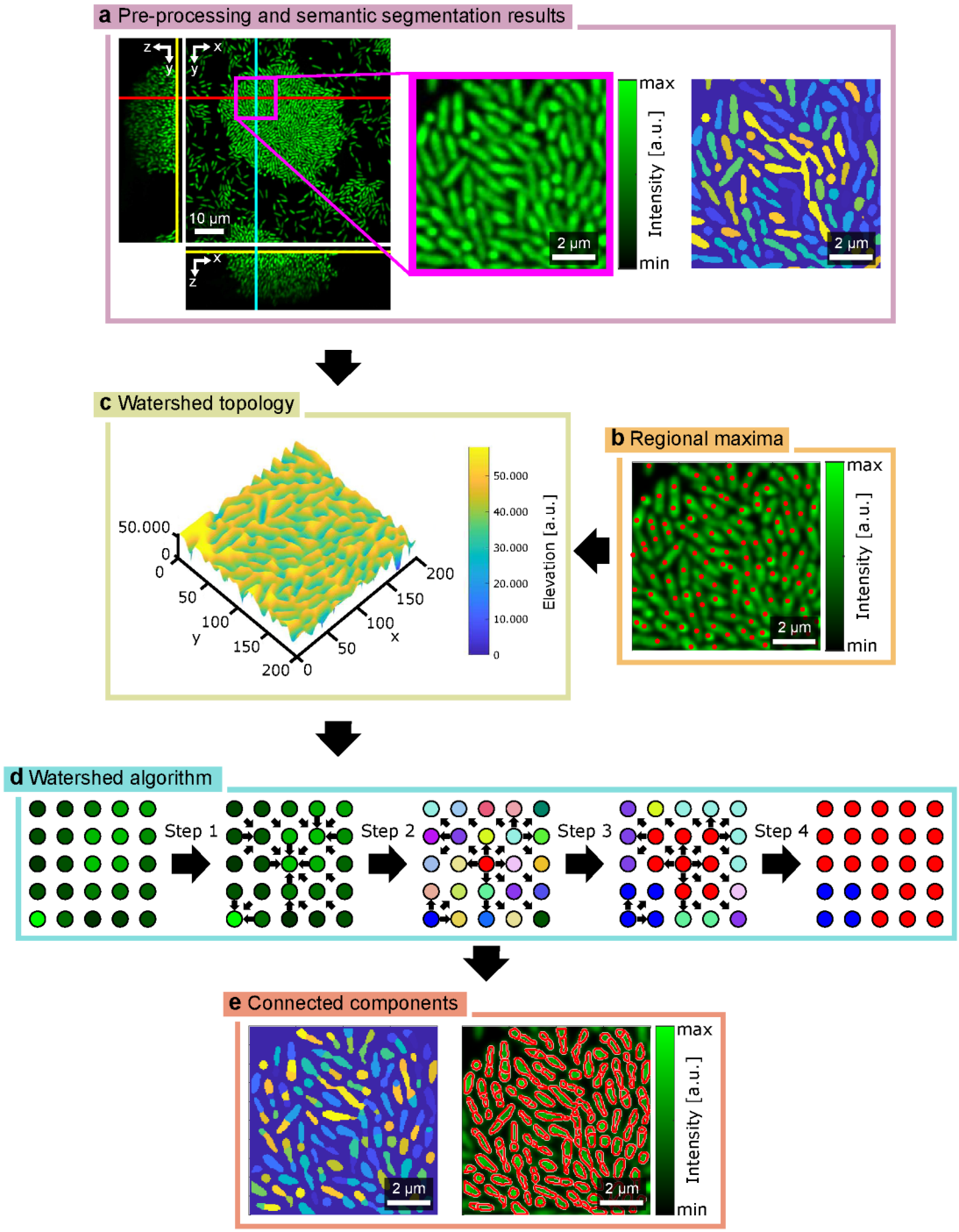


Figure 16: Watershed-based instance segmentation **a** The intensity values and the semantic segmentation are used to dissect large connected clusters. **b** Regional maxima indicate cell positions inside the segmented volume. **c** Based on the filtered fluorescent intensity and the regional maxima, a convolution with a Gaussian kernel produces the watershed topology. Cells are represented by „valleys“ and cell outlines by „ridges“. **d** Description of the implemented watershed algorithm. (Step 1) For every voxel a pointer to the largest voxel in the neighbourhood is generated. (Step 2) The pointer direction is inverted and the index of the larger voxels assigned to the neighbouring voxels. (Step 3) Repeat this step, until (Step 4) a steady-state is reached. **e** The resulting instance segmentation splits large objects into smaller sub-objects but suffers from over-segmentation.

the regional maxima by a fixed factor and a subsequent blurring with a Gaussian filter kernel. The variance of the Gaussian kernel is approximately 1.2 times larger than the user-defined cell width to propagate the information on the local maxima to the neighbouring voxels. Since the bespoke watershed algorithm searches for local minima, the resulting intensity map has to be inverted to serve as watershed topology (Fig. 16d).

To illustrate the single steps of the watershed implementation, a non-inverted pixel grid is shown in Fig. 16e. Based on the intensity values, a pointer is assigned to every single pixel (Fig. 16d, Step 1). The direction of the pointer indicates the direction of the largest gradient ascent. If no surrounding pixel has a higher intensity than the investigated pixel, no pointer is assigned. Next, the pixel intensity is exchanged by a unique identifier and the pointer direction reversed (In Fig. 16d, Step 2, the identifiers are indicated by colours). The identifiers are propagated along the pointer directions (Fig. 16d, Step 3). This step is repeated until a steady-state is reached (Fig. 16d, Step 4). Given that the intensity is highest in the cell centre and lowest in the cell perimeter, this method in combination with a previous semantic segmentation can be used as an instance segmentation (Fig. 16e, left).

On a closer inspection of the resulting instance segmentations in Fig. 16e, however, a strong over-segmentation can be observed. The morphology of the resulting objects does not necessarily correspond to the biological expected cell morphology. This is a known disadvantage of the watershed segmentation [35, 116]. Therefore, the pre-selection of the corresponding regional maxima i.e. watershed seeds is a crucial step for good watershed results.

To mitigate the observed over-segmentation and to achieve the stated high accuracy, the segmentation by Hartmann et al. employs a heuristic merging and splitting step as described in Sec. 3.6. For the post-processing procedure, an over-segmented volume is preferred over an under-segmented one due to the simplicity of a merge operation in comparison to a split operation. For merging two objects in a label image, the identifiers of both objects have to be set to the same value. For splitting, however, every single voxel has to be assigned to two or even more final object identifiers.

4.4 Accuracy Evaluation

The accuracy of the described image analysis pipeline was evaluated on five semi-manually segmented biofilms (in the following denoted by *biofilm 1* - *biofilm 5*). Information on the used *Vibrio cholerae* strain, the used SDCM parameters, details on the ground truth dataset, and the generation process can be found in Sec. 5. The used biofilms are submerged micro colonies of roughly 1000 to 6000 individual cells.

The constitutive fluorescence channel was analysed with the previously described algorithms (Sec. 4.2 and Sec.4.3) within the *BiofilmQ* framework. The relevant

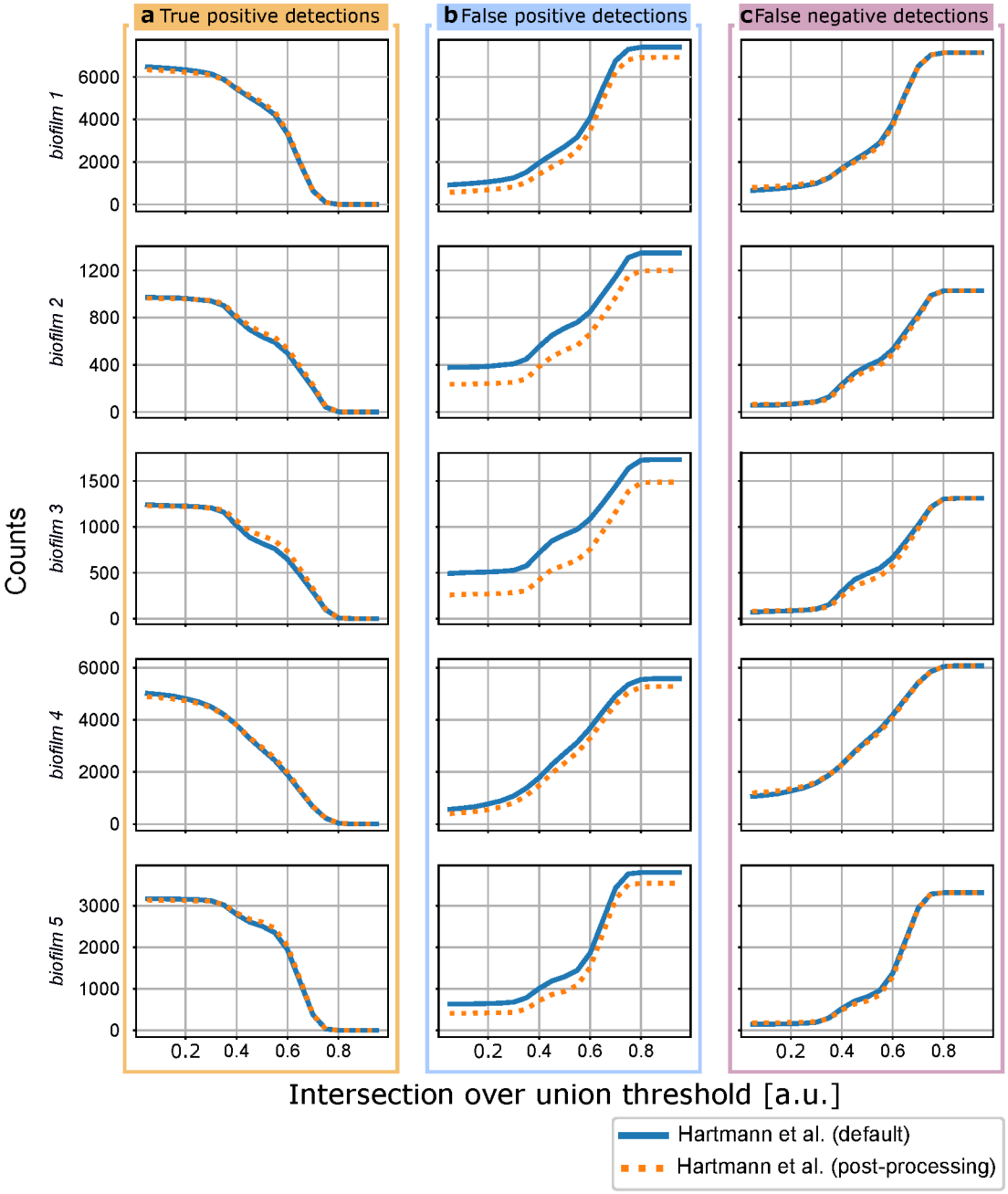


Figure 17: Segmentation results of the algorithm proposed by Hartmann et al. [66]. **a** True positive detections of the described algorithms in dependence on the IoU threshold value show only minor improvement by the suggested post-processing step. **b** The false positive detection benefits from the post-processing step. While smaller biofilms *biofilm 2*, *biofilm 3*, and *biofilm 5* (Figs. 23 & 24) benefit substantially from an additional post-processing step, the improvements are less pronounced for the larger biofilms (i.e. *biofilm 1* and *biofilm 4*). **c** The false negative detection benefits only slightly from the post-processing step.

Table 2: Semantic segmentation parameters used in the presented number accuracy calculations in Fig. 17 and overall accuracy comparison in Fig. 19.

Parameter	Value
Noise reduction kernel x & y	9 px
Noise reduction kernel z	7 px
Top hat filter kernel size	9 px
Kernel size	13 px
Gamma exponent	2
Min object voxel size	100 voxels

Table 3: Parameters of the intelligent merging and splitting heuristics used for the accuracy calculations in Fig. 17 and Fig. 19.

Parameter	Value
Keep small orphan cells	True
Convexity threshold splitting	0.7
Median volume fraction threshold (with convexity)	2
Median volume fraction threshold for splitting	3
Aspect ratio threshold for splitting	2
Median volume fraction threshold (with aspect ratio)	1.5
Merging strategy	Contact area
Median volume fraction merging	0.6

parameters for the semantic segmentation step are listed in Tab. 2. Here, the size of the Gaussian kernel for the reduction of small-scale noise was set to 9 px and 7 px in xy-direction and z-direction, respectively. The kernel size of the top hat filter to reduce background fluorescence was set to 9 px. The kernel size, i.e. cell width, was set to 13 px. This corresponds to an expected cell width of approximately 0.8 µm. The exponent to pronounce the edges in the data set was set to 2. All objects smaller than 100 voxels were excluded from the downstream analysis. The parameters were selected during multiple segmentation attempts followed by a visual inspection of the segmentation results on a small sub-volume. As soon as satisfactory results were achieved, the full volume was analysed and only minor parameter fine-tuning applied.

Additionally, the input volumes were analysed with the available splitting and merging heuristics (Sec. 3.6). The parameters for this post-processing step can be found in Tab. 3. The *Keep small orphan cells* parameter controls the processing of small cells in the data set. By setting it to *True*, cells with a volume smaller than the merging threshold remain in the data set after the final merging step. Note that these cells emerge in the splitting step during the post-processing. The remaining parameters in Tab. 3 are used to control the splitting and merging steps. In detail, three heuristic rules identify under-segmented cells and an additional rule the processing of over-segmented cells. 1) All objects with three times the median volume are unusually large for a clonal biofilm population and have to be split. 2.) Bacterial cells generally exhibit a convex shape, thus segmented object

with a convexity ratio smaller than 0.7 and a volume larger than two times the median volume do not exhibit the expected cell shape and have to be split 3.) Especially rod-shaped bacteria have a longer principal axis and two minor axes with approximately the same length. This indicates that objects with a minor axes length ratio larger than two and a volume larger than 1.5 times the median value are misshaped and under-segmented. The heuristic rule to detected over-segmented objects simply detect small cell volumes. For clonal populations, an object with a volume smaller than 0.6 times the median volume is unusually small and can be merged with an object in close proximity. If multiple candidates for merging are available, the object with the largest contact area is preferred.

The accuracy of the presented segmentation pipelines is measured by determining the TP, FP, and FN detections with respect to the intersection over union (IoU) as described in Sec. 2.7. Briefly, objects in the segmentation results are compared with the objects in the ground truth dataset by calculating the per-object overlap. For each overlap, the IoU value can be calculated. All segmented objects with an IoU larger than a set threshold are considered as TP detections [172]. Predicted objects with a smaller IoU than the defined threshold are marked as FP detections. Note, that a predicted object which does not overlap with an object in the ground truth dataset has automatically an IoU value of zero. If an object in the ground truth dataset could not be matched with a predicted object with an IoU value above the threshold, it is counted as a FN detection.

The IoU-dependent TP, FP, and FN detections for the implemented segmentation pipelines as defined by Hartmann et al. are shown in Fig. 17. To distinguish the differences between the test sets with respect to the biofilm size, the different detection values are shown in separate diagrams.

The TP detections – i.e. the number of objects in the ground truth data which are correctly recognized by the segmentation pipelines – are shown in Fig. 17a. The TP detection numbers close to an IoU of 0 indicate the maximal number of overlapping objects between ground truth data and segmentation results. It is to be noted, that an algorithm with one-to-one matching is used [172]. Therefore, an over-segmented ground truth object is only counted exactly once as a TP detection if the IoU criterion is met. If the IoU threshold becomes larger, the segmented objects have to match the object shapes in the ground truth more closely. In the extreme case of an exact replicate of the ground truth data by the segmentation pipeline, no dependence of the TP curve with respect to the IoU would be observable.

In current segmentation pipelines, however, the number of TP detections decrease with an increased IoU threshold. The slope of the TP curves indicates how different the segmented objects are from the ground truth labels created ones. A steep slope indicates that many objects have a similar volumetric segmentation error. For all tests (except *biofilm 4*) a turning point can be observed. This turning point indicate different classes of segmentation errors in the dataset were one class better replicates the ground truth objects than the other.

In the presented segmentations, the post-processing heuristics only resulted in

a very small increase in the TP detections for the small biofilms *biofilm 2* and *biofilm 3*. For larger biofilms (i.e. more than 2000 individual cells), no effect of the post-processing step on the TP detection could be measured.

If no matching ground truth object for an object in the segmentation result can be found, the object is marked as FP. The post-processing approach introduced by Hartmann et al. [66] had measurable effects on the corresponding detections (Fig. 17b). For all investigated datasets, the number of FP detection decreased by a constant offset. The height of the offset depended on the individual dataset. For *biofilm 3* the post-processing resulted in approximately 250 less FP detections, while for *biofilm 4* a decrease of approximately 50 FP detections could be measured. This offset is connected to the successfully merging of over-segmented cells.

Similar to the effect on the TP, the post-processing step does not have a measurable effect on the FN detections in most investigated datasets. Only the FN detection curve for *biofilm 3* shows slight variations between the default segmentation and a segmentation with heuristic post-processing. Further, the relative segmentation errors can be compared: The number of FN detections at an IoU of 1 directly show the number of objects in the ground truth dataset. The FN detections at an IoU of 0 shows the number of ground truth objects which are not captured by the segmentation pipeline. Here, *biofilm 4* shows a large number of FN detections. Approximately 20% of all objects could not be detected with the chosen parameter settings for the segmentation pipeline. On the other tests sets (*biofilm 1*, *biofilm 2*, *biofilm 3*, and *biofilm 5*) the number of not detected ground truth objects is below 10%. This is comparable to the estimate by Hartmann et al.

Overall, the post-processing step improved the relative FP detection rate on smaller biofilms more than in larger biofilms. Yet, the number of FP detections remains high. Over-segmented cells are the major source of segmentation errors in the datasets.

4.5 Improving Segmentation Accuracy

As shown in the previous section, the over-segmentation and more precisely the large number of FP detections limits the average precision of the image segmentation pipeline proposed by Hartmann et al. The key driver for the large number of over-segmented objects is a liberal estimation of watershed seeds by the regional maxima calculation. For this reason, a more restrictive selection of the regional maxima can benefit the segmentation accuracy and improve the feature extraction of the following downstream analysis pipeline. To improve the segmentation, maxima within the same cell have to be merged.

A schematic explanation of such a seed-selection algorithm is shown in Fig. 18. For illustrative purposes, the approach is shown in 2D. The implementation can be extended to 3D volumes without loss of generality.

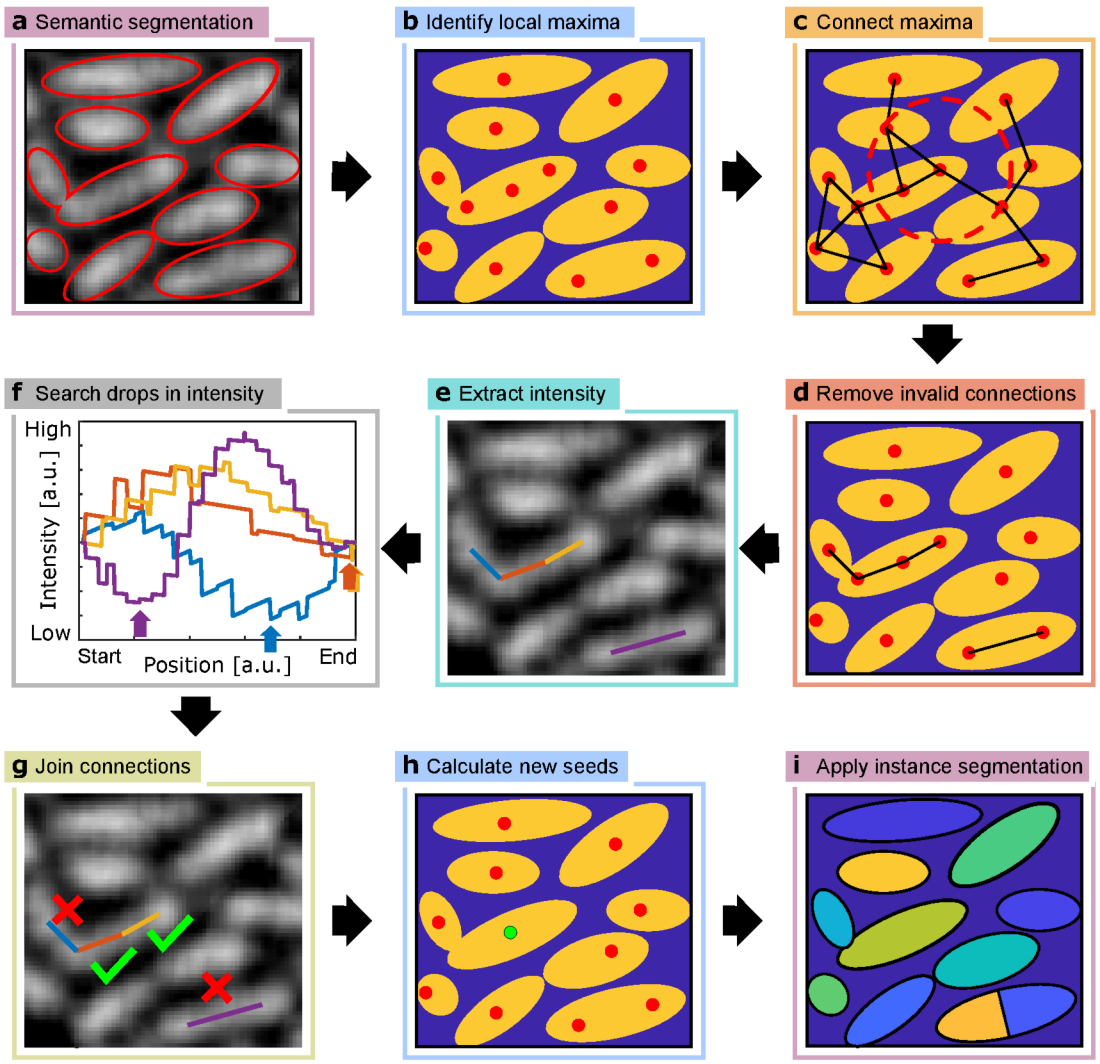


Figure 18: Schematic overview of the proposed instance segmentation. **a** In a previous semantic segmentation step the foreground voxels are detected. **b** Based on the available fluorescent intensity values, local maxima can be identified. **c** The distance from a single maxima to all other available maxima are calculated. If the distance is lower than a user-defined search radius, a connection between the maxima is established. **d** All connections which cross background voxels are removed from the connection network. **e** The voxel intensity along the connection is extracted for every single connection. **f** In the normalized intensity profile global minima are extracted. **g** Based on Otsu's method on the global minima values connected and disconnected maxima are extracted. **h** Connected maxima are pooled together into a single seed. **i** A seeded watershed is used as an instance segmentation.

Table 4: Parameters of seeded watershed instance segmentation as used for the accuracy calculations in Fig. 19.

Parameter	**Value**
Cell number threshold	10
Max connection distance	100

The basis for the seed-selection algorithm is an available noise-reduced fluorescent signal and a corresponding semantic segmentation similar to the edge-detection-based segmentation presented in Sec. 4.2. In principle an accurate threshold-based segmentation is sufficient (i.e. the approaches described in Sec. 3). This simplifies the integration of the new algorithm into the *BiofilmQ* framework and enables a quick adaptation in upcoming publications. Based on the semantic segmentation and the local maxima (Fig. 18b), a network between the single maxima can be constructed (Fig. 18c).

To reduce the compute time of the analysis, the number of connections can be limited by using a search radius around each maximum (indicated by the red circle in Fig. 18c). Only maxima within the search radius are connected. For each connection between the maxima, the voxels along the connection line are checked. As soon as one of the voxels belongs to the background class of the semantic segmentation, the connection is omitted (Fig. 18d). This results in a huge reduction of the possible maxima connections and benefits the processing speed.

For the remaining connections, the intensity of the voxels along the connection line is extracted. The resulting intensity profile is normalized such that the start- and end-point of the profile – both of which are local maxima in the blurred intensity volume – are equal to zero. All intensity values between those points are updated correspondingly. The minimal value of each profile is extracted (indicated by arrows in Fig. 18f). For a sufficient large number of representative connections, Otsu's method separates the minimal values into two classes: Ones that exhibit a low value and ones that do not. While the former indicates that the two connected maxima belong to two different cells (i.e. separated by background fluorescence), the latter gives rise to the assumption that the two maxima belong to the same cell and should be merged (Fig. 18g & h). After this step, a regular watershed algorithm can separate the objects into single cell instances.

With this extension, *biofilm 1 - 5* can be resegmented and compared with the ground truth data. For a fair comparison, the same edge-detection-based semantic segmentation is employed and the same local maxima as depicted in Fig. 17 are used. The parameters of the new instance segmentation algorithm are listed in Tab. 4. The parameter *Cell number threshold* defines the minimal number of maxima which are required for the application of the presented method. Here, the initially available number of regional maxima is two orders of magnitude larger than the defined threshold value. This is a safety mechanism for small biofilms with a small number of individual cells. With a low number of maxima, the automatic threshold estimation might produce inconsistent results. The parameter

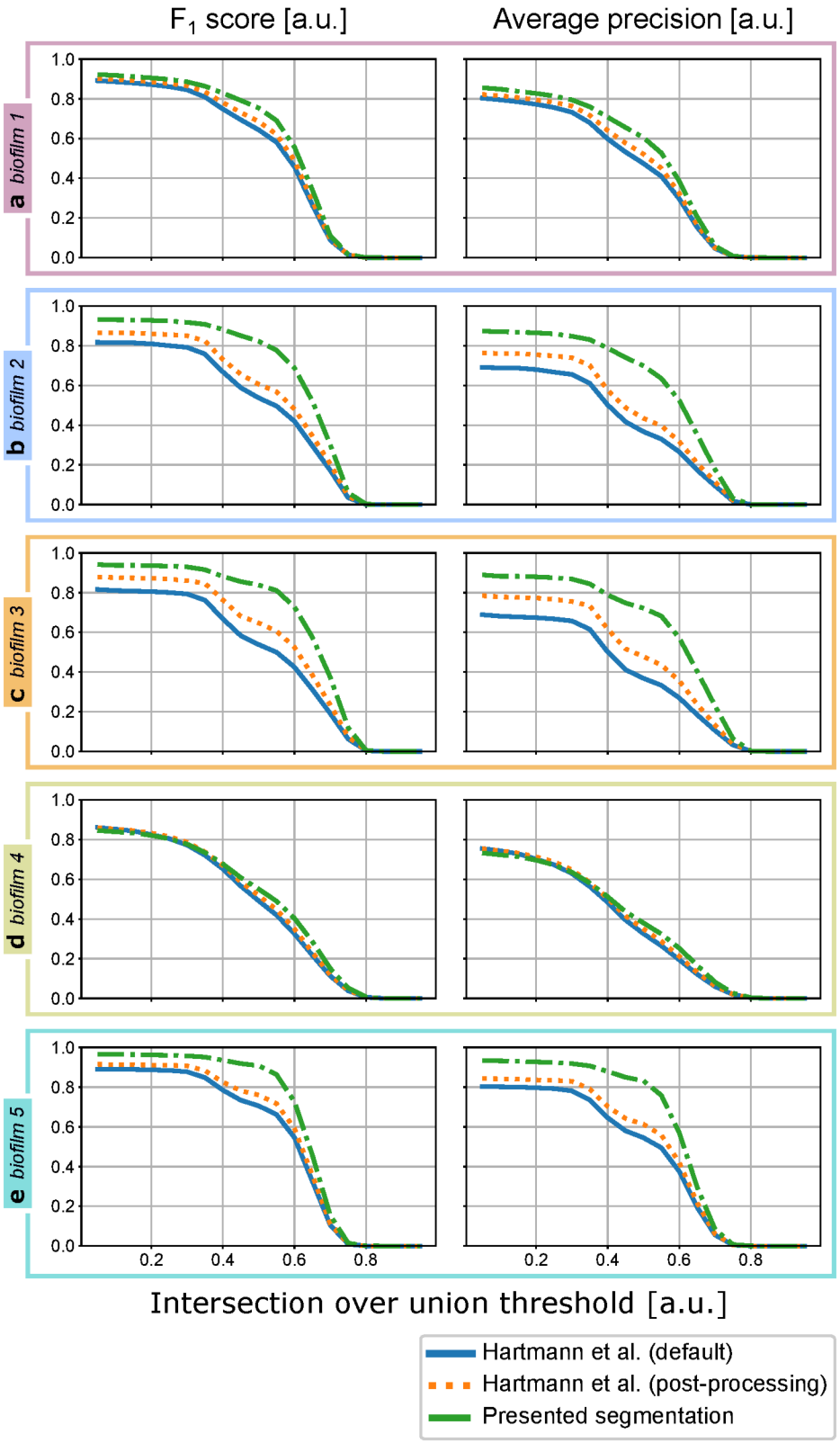


Figure 19: Accuracy comparison of classical segmentation algorithms for different test datasets. **a** Large biofilm (*biofilm 1*). **b** & **c** Test sets of small biofilms (*biofilm 2* and *biofilm 3*). **d** Test biofilm with small number of single cells attached to substrate (*biofilm 4*). **e** Second test set with a large biofilm (*biofilm 5*).

Max connection distance defines the maximal pixel length of candidate connections between maxima. A smaller value limits the maximal distance between maxima investigated for merging. In comparison to the required parameters for the heuristic merging and splitting (Tab. 3), the given input parameters do not directly affect the volumes of the resulting objects. Instead, the parameters are used to prevent long processing times and misleading results in the automatic threshold step.

The resulting F_1 score and average precision for Hartmann et al. with and without post-processing, and for the proposed instance segmentation approach are shown in Fig. 19. For all investigated test sets, the accuracy improvement in both metrics is significant. In general, the post-processing employed by Hartmann et al. benefits the segmentation accuracy. Only for *biofilm 4* the additional post-processing step does not improve the accuracy metrics. The measured accuracies at the commonly used IoU reference threshold of 0.5 does not exceed values above 0.6.

The newly proposed instance segmentation strengthens the segmentation accuracy of edge-detection-based semantic segmentations. All metric values are well above the ones achieved with the current segmentation used by Hartmann et al. and Zhang et al. [183] (private communication). On some datasets (i.e. *biofilm 2*, *biofilm 3*, *biofilm 5*) the algorithm reaches very good segmentation results. The performance on the large biofilms (i.e. *biofilm 1* and *biofilm 4*), however, is still not usable for automatic lineage tracking inside bacterial biofilms. Surprisingly, the performance on *biofilm 4* benefits neither from the available post-processing pipeline, nor from a more restrictive selection of the watershed seeds. One reason for this could be a misleading threshold in the semantic segmentation. An incorrect semantic segmentation cannot be corrected by a downstream instance segmentations without major changes to the *BiofilmQ* framework.

4.6 Conclusion

In this chapter, the performance quantification of a current classical segmentation approach for volumetric microbial biofilm images has been presented [66]. The previously stated number accuracies of 95% could be replicated on the annotated test sets. This number accuracy, however, is insufficient to describe the per-voxel segmentation accuracy. Only with a per-voxel-based segmentation, single-cell features such as shape, volume and orientation can be described accurately.

Based on the insights gained on the TP, FP, and FN measurements, I quantified the influence of a heuristic merging and splitting post-processing procedure, especially for the reduction of FP detections which arise due to over-segmented cells. Heuristic merging and splitting decreased the FP numbers, but could not completely remove wrong detections from the segmentation results. Note, that the manual determination of the optimal parameters for the heuristic thresholds is bound to a potential segmentation bias. Yet, a full parameter scan is often impractical in phenotype-centric publications due to *a priori* missing ground truth

data.

Instead of further optimizing the available post-processing pipelines, I proposed an extension to the watershed-based instance segmentation with a reduced number of free parameters. The remaining parameters either can be extracted from bright field observation of single cells (i.e. the maximal observable length of a single cell) or are safety settings to prevent a misleading threshold estimation for small biofilms. Note that even the semantic segmentation performs remarkably well if only a small number of cells need to be analysed. The proposed extension reduces the over-segmentation in the dataset by effectively reducing the number of watershed seeds – the main reason for the previously observed over-segmentation.

Based on the accuracy measures, the segmentation accuracy could be improved for all evaluated datasets at the relevant IoU value of 0.5. In some cases, the algorithm exhibits an average accuracy above 0.8. On some other datasets, however, the accuracy score indicates non-satisfactory results for all investigated segmentation pipelines.

A dataset-dependent bias cannot be completely excluded. However, the test datasets were not specifically designed for the accuracy measurements of classical segmentation pipelines. Instead, all datasets with available ground truth segmentation were used. The results indicate that further improvement can be achieved. Yet, all classical pipelines suffer from poor generalization results and have to be optimized for each individual experiment [160]. This fine-tuning for each experiment can be avoided by using learning algorithms. Learning algorithms already dominate current cell segmentation competitions and will slowly replace most of the non-learning-based segmentation pipelines in publications [160].

The presented seed reduction algorithm can be further improved by direct learning of the watershed map [8, 176]. In the current implementation of the segmentation pipelines, temporal information is not used for the segmentation procedure. By pooling the information of multiple frames together, an additional post-processing step can use further heuristics to improve the segmentation accuracy [6, 113]. Every single object in the over-segmented results can be used as a super-voxel which are merged and rearranged into objects which remain coherent across time frames. The incorporation of super-voxel based heuristics would require a complete refactoring of the *BiofilmQ* framework.

To improve the segmentation accuracy further, the number of FN has to be decreased. This requires new approaches for the semantic segmentation. Implementing such a pipeline is a risky endeavour for an output-based research environment, since the segmentation accuracy improvement can not be guaranteed. The labour for manual feature design and reprogramming a segmentation pipeline is probably better invested in generating ground truth data for supervised learning algorithms [118].

First deep learning-based segmentation approaches emerged recently [139, 162, 48, 172, 156], and rely on accurate ground truth data annotations [118]. A

possible path to circumvent this problem and leveraging the powerful *Stardist* [172] and *Cellpose* [156] implementations can be found in Sec. 5. For label-free segmentations [163], however, robust classical algorithms will remain the preferred choice for quick analysis of standard microscope experiments.

5 Deep Learning-based Segmentation in Biofilms

As seen in the previous section (Sec. 4), the performance of current classical segmentation pipelines for biofilm segmentations – while already suitable for a wide array of analysis requirements – is still far from perfect. To improve the accuracy further, the semantic segmentation as well as the instance segmentation step needs to be rewritten. Even with new feature extractions, the algorithms would need fine-tuning for different imaging conditions and strain properties. A successful reimplementation of a suitable segmentation pipeline would require months of tedious programming.

5.1 Introduction

To circumvent the reimplementation of new custom segmentation pipelines, it is beneficial to step back and review the currently best performing algorithms in the field. The field has dramatically changed in the last decade. The currently available algorithms can be divided into three classes: 1) Classic image processing algorithms 2) Traditional machine learning algorithms, and 3) deep learning methods [26].

Classical segmentation pipelines require manually designed feature extractions in the shape of edge-detection filters, thresholds, and additional preprocessing steps. In contrast, traditional machine learning algorithms (i.e. random-forest [76] or support vector machines [32]) treat the feature extraction as an optimization problem, where input and output data are given as already extracted features, and a model is fitted to the available data to „learn“ the connection between input and output. Example implementations of such algorithms for the segmentation of biological datasets can be found in the *Trainable WEKA segmentation* [7] or *Illastik* [16, 152].

Traditional machine learning pipelines offer very fast runtimes, but cannot be trained end-to-end. This means, that always handcrafted feature extractions need to be implemented prior to the model optimization. Deep learning algorithms, however, can learn representations of the features from raw data and do not require manual feature extractions [98]. Enabled by the progress in compute capabilities of current GPU hardware, deep learning can extract a rich feature sets from image data input [27].

In the field of object detection, deep learning produces state-of-the-art results [98] and became a standard tool in biological image processing [118]. The internal representation of the input spaces does not only capture single values, but extracts the spatial context for a pixel – similar to the user-defined filter approaches in the classical image analysis [48]. The filter kernels are learned implicitly during the optimization steps (commonly termed „training“). Since the first successes in image analysis competitions, the performance steadily increased and surpassed

even the most elaborated classical image segmentation pipelines [160].

The success of deep learning for image processing algorithms started with the seminal paper by Krizhevsky et al. [95]. By using a neural network with 60 million free parameters, Krizhevsky et al. achieved record-breaking results for image classification on the ImageNet dataset [95, 49]. Even though these classification results started a new era in the application of deep learning algorithms, I consider the NIPS 2012 contribution by Cireşan et al. as more important for the bioimage community. It can be viewed as the first application of pixel classifier to biological image data in the shape of challenging electron microscopy (EM) outputs [29].

The deployment of fully convolutional neural networks (FCN) [107], in combination with sophisticated data augmentation [147], autoencoder [74], and the idea of skip connections enabled the creators of *U-Net* [139] to win the ISBI 2015 cell tracking challenge [160] by a large margin. *U-Net* became quickly popular in the biological image processing community due to its moderate training data requirements and adaptability to 3D datasets [28, 48].

The successes of image classifiers and the *U-Net* architecture were entirely built on their superior performance in common image analysis challenges. A first attempt to simplify the application of the algorithms was the publication of *DeepCell*. It measured the real world performance of pixel-wise classifications for instance segmentations in fluorescent microscope images [162]. Due to the pixel-by-pixel prediction, *DeepCell* is a computational costly algorithm [29] and uses active contour post-processing to extract the single object segmentations from the deep learning prediction results [162].

Currently, most of the newly presented segmentation networks use residual network [67] or *U-Net* [139, 28, 48] architectures as backbones and add custom network heads for the task-specific adaptation. The residual network backbones have been subject to major improvements manifested in the shape of *Fast R-CNN* [54], *faster R-CNN* [136], *Mask R-Net* [69] in recent years, while the *U-Net* architecture remained almost unchanged for the application of single cell segmentation. Today, neural networks for biological image segmentation are considered superior to classical segmentation pipelines [160]. One reason is the reduced number of biologically relevant segmentation errors (i.e. over-segmentation, under-segmentation or missing objects) [22].

Recently, networks with encoded shape models were introduced to the bio-image analysis community [143, 172, 156, 46]. Encoding shape-related priors into the training data enables networks to directly learn shape-related features from the training samples. This allows for direct object detection on the network outputs without additional post-processing steps for the instance segmentation. The most popular implementations, namely *Stardist* [143, 172] and *Cellpose* [156], come with a well written online documentation and can be quickly adapted to custom 2D datasets. Both build on the already mentioned backbones and apply their individual shape models by using sophisticated network heads. While *Stardist* is designed for the segmentation of star-convex nuclei in cell tissues, *Cellpose* was created to correctly segment different cell shapes under diverse imaging

conditions. Based on the stated segmentation accuracies, a large body of publications will use these models for the segmentation of challenging datasets with densely packed cells in the near future.

At least as important as ever-higher segmentation accuracies from more sophisticated models are the training datasets. Currently, no large collection of volumetric biofilm image data for testing and training of machine learning algorithms exists. In the case of 2D segmentations, the 2018 Data Science Bowl dataset [23] enabled researchers in the field of computer vision to test their algorithms in real world scenarios.

Annotation of 3D data is tedious and requires a large investment of manual labour to reach a suitable training set size. Currently, the number of research groups which use machine learning approaches for the segmentation of biofilm data are limited [183, 181]. Only with the availability of high quality training data, progress in the field of single-cell segmentation inside bacterial biofilms is likely to emerge in the future.

The lack of suitable training data becomes more serious from the performance perspective. With some limitations, the accuracy of learning algorithms scales with the abundance of training data [63, 158]. Large quantities are invaluable to avoid overfitting and to improve the generalization capabilities of the used networks. A similar effect is likely to emerge on the field of single-cell segmentations inside bacterial biofilms. Therefore, not only the existence of publicly available training data but also its abundance becomes critical to refine the segmentation of single cells inside dense microbial communities.

In the following, I investigate the accuracy of an end-to-end application of the two promising deep learning segmentation algorithms *Stardist* [172] and *Cellpose* [156] on the field of single-cell segmentation in biofilm research. To address the absence of training data I present a simple, yet effective iterative learning approach. The approach is suitable to quickly generate annotated 3D biofilm datasets. Furthermore, I test how the deep learning approaches compare with current classical segmentation pipelines and how large quantities of training data influence the segmentation performance of the best-performing algorithm.

5.2 Related Work

First applications of deep learning based segmentations of biofilms or dense microbial communities are emerging [183, 181]. In these studies, *U-Net* architectures were used for semantic segmentation. The authors relied on their custom post-processing pipelines for the instance segmentation. The post-processing either used custom graph-cut techniques [168, 183] or the common watershed segmentation [181] to reach the stated values in IoU or average precision. The weakness of custom post-processing pipelines is the separation of the post-processing from the learning algorithm. This separation prevents the algorithm to directly learn object features for the instance segmentation end-to-end from the training samples and cuts off the scaling benefits of data abundance from the post-processing

algorithms. In contrast, *Stardist* and *Cellpose* incorporate the instance segmentation information directly in the output layer of the network.

The possibility to improve the performance with more abundant data is an undeniable benefit of machine learning algorithms for image analysis tasks [158]. Thus, the availability of training data is the key driver on the field. Parallel to the attempts to source labelled image data for object detection and image classification tasks on macroscopic scales [99, 94, 49, 103], the 2018 Data Science Bowl build an extensive set of annotated fluorescence microscope image data [23].

One way to extend the available 2D annotation data to volumetric annotation is the application of sparse annotation [28, 48]. Instead of annotating a full volume, only selected planes are annotated. By training an *U-Net*, the full volume can be predicted and the labelling effort is dramatically reduced. *Cellpose* employs a similar concept where only 2D training data is needed. By predicting the direction to the cell centre, a simple vector addition leads from 2D annotation data to 3D predictions. In the case of 3D biofilms imaged with SDCMs, the resolution in the axial direction is well below the planar resolution and thus very challenging to annotate. Additionally, the effects of the PSF is enhanced in the axial direction, which results in distorted cell shapes [66].

Instead of reverting to 2D data, many deep learning-based algorithms employ simulated 3D data for the training process [183, 42, 75]. Simulated data provide the benefit of the easy accessible label data. Real microscope data, however, do not exactly exhibit the simulated noise distributions. This is why, simulated data does not one-to-one replicate the physically present imaging conditions emerging during the observation of bacterial biofilms with a fluorescent microscope. Networks trained on simulated data do not perform well on experimental image data [123].

Besides learning style transfer to reuse existing expert annotated datasets under different imaging conditions [77] or using threshold approaches to use a fluorescent channel as annotation label for bright field images [140], iterative annotation approaches seem to be one path to avoid manual annotation of 3D datasets and have been implemented for eukaryotic systems [26, 177]. In an iterative annotation, an expert validates intermediate results and thus improves the annotation accuracy [19, 2].

The refinement of already trained networks has also been employed with the *U-Net* architecture to reduce the annotation burden in a medical setting where expert annotations are expensive [86]. This can further be improved by active learning approaches. Active learning tries to select the training samples for manual annotation which reduces the segmentation error the most. Simple implementations calculate uncertainty scores for full predictions and incorporate the data into the training data depending on the score [180, 169]. For a full biofilm volume this is not a viable option, since even a moderate-sized single biofilm consists of 1000 to 6000 3D objects. The amount of objects makes the annotation from scratch infeasible for small research groups. Therefore, a sub-volume based labelling approach is needed to correct segmentations. Active learning

sample selection mechanisms already provide proof-of-concept implementations and showed that the number of annotated training samples can be reduced, while achieving similar (or even higher) classification accuracies [90, 91, 184].

Currently, it is unclear how many individual objects are necessary to successfully segment dense microbial communities. For sparse cell distributions in 3D, 14 training stacks with a size of ($236 \times 236 \times 100$) resulted in an IoU value for a semantic segmentation of 0.8 [48]. For 2D data, even a single image with approximately 300 bacterial cells was sufficient to create a suitable pixel classifier for single-cell segmentation [162]. In the original *U-Net* publication 2D segmentation is feasible with as low as 30 images [139]. In follow-up publications, the same research group reported that even 2-10 images might be sufficient to converge to reasonable results [48].

High segmentation accuracies with a low number of training samples can only be achieved with heavy data augmentation and the extraction of sub-volumes at random locations inside the image [147]. In the case of *DeepCell*, data augmentation provided 200.000 - 400.000 sub-image patches per image by exploiting the system symmetries and using small patch sizes of 32×32 pixels [162].

My contributions to the field are the following: 1) Implementing a pipeline for the quick ground truth data annotation. 2) Adapting two end-to-end deep learning pipelines to the segmentation of bacterial biofilm image analysis. 3) Benchmarking the pipelines average precision with respect to the currently existing bacterial biofilm segmentation pipeline in *BiofilmQ*. 4) Qualitatively describe typical segmentation errors in the datasets. 5) Quantify the needed amount of training data to reach robust segmentation results.

5.3 Method

Iterative data annotation To quickly generate large amounts of training data, an iterative approach was chosen (Fig. 20). First, a small dataset was manually generated without using deep learning-based segmentations. The preliminary training samples were reviewed for errors. In a following step, a human annotator corrected the errors manually. This generated a small training set, which was used for the optimization of a randomly initialized deep learning algorithm. To improve the overall performance and increase the variance in the available training data, data augmentations such as random flips, rotations, and artificial signal level shifts were introduced during the training of a new segmentation model. The trained segmentation model was applied to yet unknown microscope data. The results of the prediction were used as new label predictions. The label predictions were checked for segmentation errors based on an automatized metric. The errors were either automatically corrected or in a second manual annotation session corrected. This produces an iterative training scheme of constantly improving and growing training data.

The outcome is a segmentation model, which 1) is trained on a large training dataset and 2) generalizes well on new input data. By automatizing the error

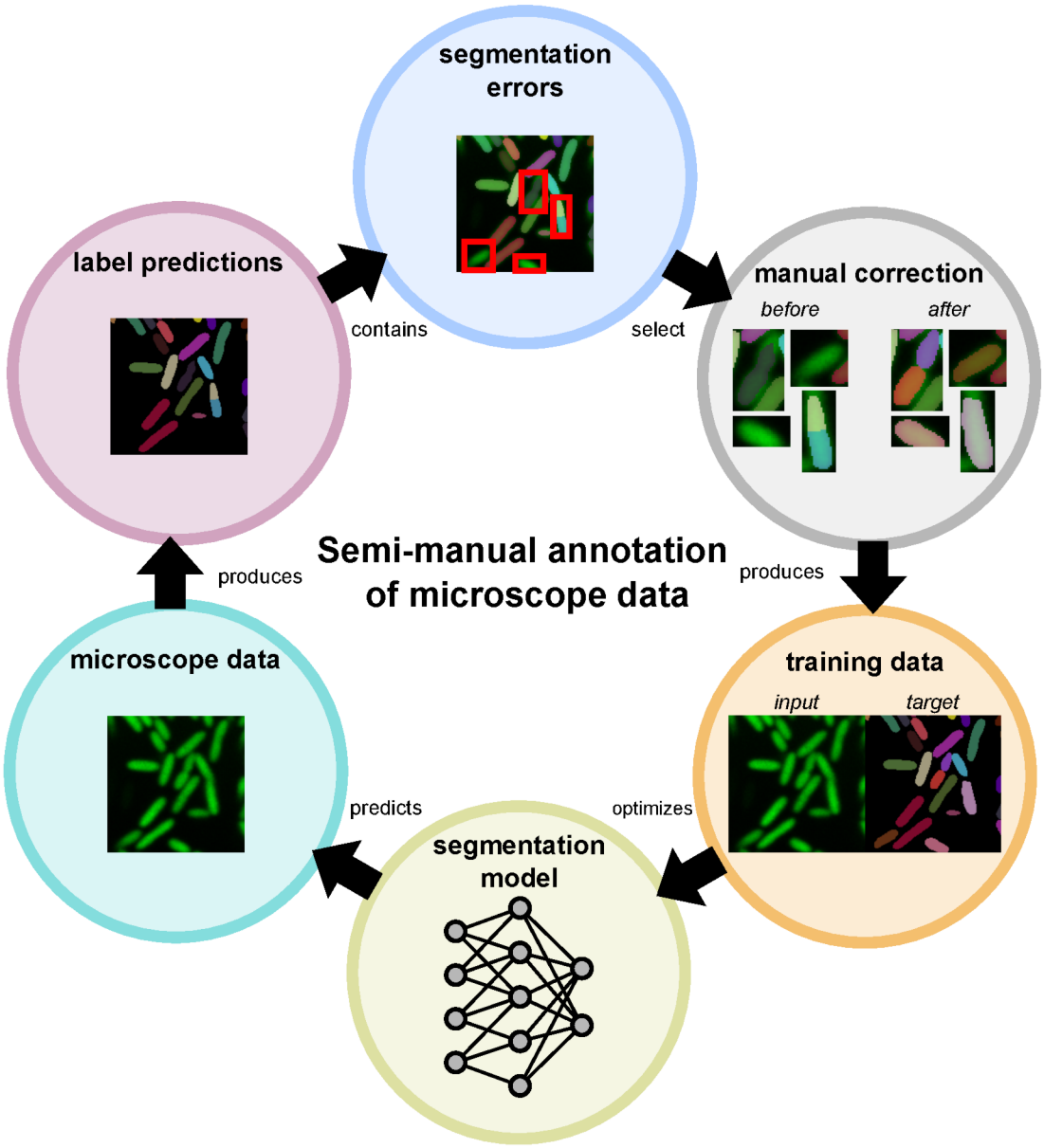


Figure 20: Overview iterative annotation of training data. Label predictions are produced from collected microscope data either manually or with a segmentation algorithm. The segmentation errors are identified either manually or with an automatic error detection. Based on the error detection, segmentation errors can be corrected individually without reviewing the complete dataset. The corrected labels and the microscope data are combined into an error-free training dataset. The dataset is used for the training of a segmentation model. The model in turn is applied to annotated microscope data and produces a label prediction. Again, the remaining segmentation errors are corrected and a more accurate segmentation model is trained.

detection and stream-lining the manual correction, the required time for the annotation is minimized while the training data output is maximized.

The initial error selection and the manual correction in Fig. 20 requires a large time investment. To automatize the error selection, a custom error selection algorithm and a GUI for the error classification were implemented in *MatLab*. The model produced a large set of object suggestions as depicted in Fig. 21a. Based on the object suggestions, voxels with conflicting predictions could be spotted and extracted one-by-one from the dataset Fig. 21b. Instead of correcting the conflicts manually (i.e. by directly using a graphical annotation tool such as *napari* [31]), a sliced representation in the xy-, xz-, and yz-plane is presented in the GUI (Fig. 21c). For each conflict, the user can decide between merging, splitting, or marking the conflict for later manual correction. This speeds up the error review process and allows for a high-throughput error correction.

Conflicts, which were marked for manual correction or which could not be corrected automatically, are loaded into *napari* (Fig. 21d). All labels without merge conflicts are hidden such that only erroneous labels appear in the interface. With additional, custom-written *Python*-based help functions [161], the annotator automatically iterates through the positions. This saves additional time during the conflict location, prevents redundant corrections, and reduces the possibility that conflict are skipped during the correction. After the manual correction, the resulting labels are merged with the automatically corrected labels. In the merged volume, all cells are labelled correctly.

For the deep learning segmentation task, two recent model architectures are tested. Both already showed superior performance on densely packed microbial communities and are fully compatible with 3D segmentation tasks. Yet, the segmentation accuracy on biofilms remains unknown. Therefore, the generated densely annotated volumes are used to train the selected architectures and for the evaluation of the segmentation accuracy.

Stardist Based on the powerful 3D *U-Net* [28] or *ResNet* [67] backbone, a *Stardist* network returns a prediction for the normalized Euclidean distance transformation for each voxel in the volume [143, 172]. The Euclidean distance transformation describes the distance from a voxel inside an annotated object to the next non-object voxel. The prediction of a maximum-normalised Euclidean distance transformation can be interpreted as a probability that a certain pixel is the centre of a yet unidentified object. *Stardist* additionally predicts the distance of each voxel to the possible cell boundary in initially defined directions. Based on these two properties, a non-maximum suppression (NMS) can create segmentations for objects in the input volume. Due to the nature of the predicted values, the predicted objects are restricted to star-convex shapes. The Euclidean distance transformation as well as the distance measures of the labelled data are calculated during the network training with highly efficient C++ or GPU-accelerated *OpenCL* implementations [88]. This enables the usage of regular object labels. No additional feature calculations are required prior to the training process. The

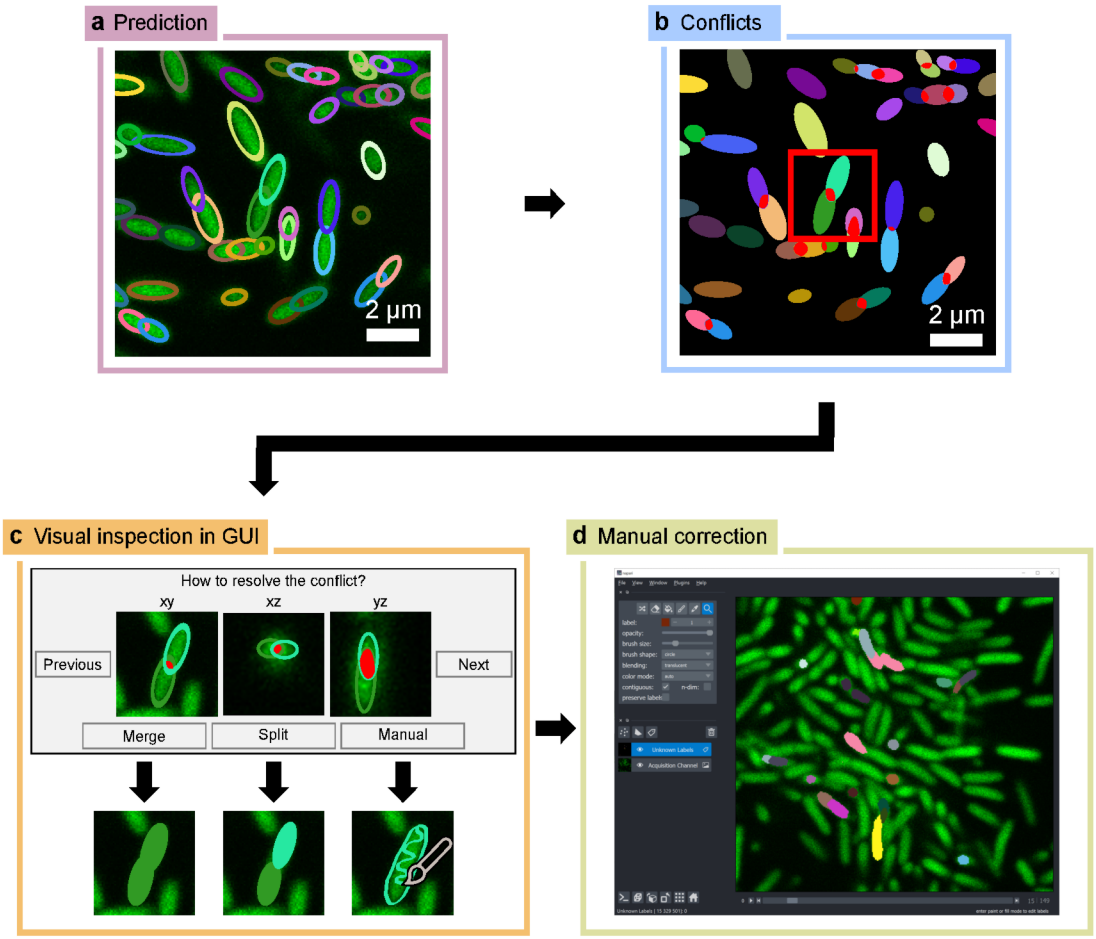


Figure 21: Overview of automatic error detection and semi-manual dataset correction. **a** Based on the microscope input, the deep learning segmentation model produces object suggestions. **b** Segmentation conflicts can be identified by overlapping segmentation suggestions. **c** Each conflict is investigated in a simple graphical user interface (GUI) for a quick inspection. The GUI supports custom key-bindings to quickly decide between the different options: *Merge* (automatically merge the two conflicting labels), *Split* (automatically split the conflicting labels), or *Manual* (mark the labels for later correction in *Napari*). **d** *Napari* interface for manual label corrections. After the correction, the labels are merged again into a single label file.

number (and thus the directions) of the distance measures are defined during the network initialization by a user-defined hyper-parameter choice [172].

Cellpose The architecture of the *Cellpose* [156] model consists of a heavily modified 2D *U-Net* [139]. To account for non-blob-like cell shapes, *Cellpose* uses a simulated diffusion model to calculate flows which point to the centre of each cell. Similar to the original *U-Net* implementation, it calculates for each voxel an object probability and additional horizontal and vertical diffusion flow [156]. Since the flow always points to the cell centre, even touching object predictions can be separated in a simple post-processing step. To extend the 2D flow information into 3D, the flows in the different planes (xy, xz, and yz) are combined to a single flow field by simple vector addition. This enables the training of a 3D segmentation algorithm with 2D training data.

5.4 Experiments

Images of rugose *V. cholerae* N16961 biofilms with straight cell shapes were used to test the segmentation pipeline [115, 66]. The cell shape emerges due to the deletion of the *crvA* gene [14]. The rugose phenotype is introduced by a single nucleotide mutation in the *vpvC* gene, leading to increased production levels of the second messenger cyclic di-GMP [18]. To render the cells visible with SDCM a pNUT542 plasmid was introduced into the cells which results in a constitutive superfolder green fluorescent protein (sfGFP) production [151].

Sample preparation *V. cholerae* was grown at 25 °C in LB medium under shaking conditions at 250 rpm. After 18 h, the overnight culture reached an optical density (OD)(at 600 nm) of 0.5. The culture was diluted 1:1000 in M9 medium (M9 minimal medium supplemented with 2 mM $MgSO_4$, 100 mM $CaCl_2$, minimum essential medium (MEM) [44], 0.5 % glucose, and 15 mM triethanolamine (pH 7.1)) and inoculated in glass-bonded Polydimethylsiloxan (PDMS) flow chambers with the dimensions 500 µm×60 µm×7 mm (width × height × length) [41].

To allow for cell attachment to the glass substrate, the flow chambers were inoculated an additional hour at room temperature (25 °C). For the next 20 h, the cells grew at room temperature with a constant M9 medium flow of 0.5 µL/min provided by a high-precision syringe pump (Pico Plus, Harvard Apparatus). Antibiotics (trimethoprim and tetracycline) stopped the cell growth and cell division 10 minutes prior to the image acquisition.

Image acquisition The single biofilms were imaged with a Yokogawa CSU-W1 confocal spinning disk unit attached to a Nikon Ti-E inverted microscope. A silicone immersion oil (with a refractive index of 1.406) objective (Olympus) with 100× magnification and a NA of 1.35 was mounted on a custom adapter to enable high-resolution imaging. An additional 2× magnification lens, which

was mounted between spinning disk unit and camera, increased the microscope magnification. The described setup images a 61 nm × 61 nm area, onto a single pixel on the camera sensor. The images were captured with an Andor iXon 888 Ultra EMCCD cooled to -70 °C. During the image acquisition, an electron multiplier gain was used. The camera readout rate was set to 10.000 MHz and a 90 ms exposure time set.

The fluorophores in the sample were excited with a 488 nm laser (OBIS) at a laser power of approximately 2 mW. To decrease the acquisition time of volumetric samples, a piezo-stage (MS-2000 FT, ASI) was used. The sampling distance in z-direction was 100 nm.

Deconvolution Prior to the data annotation, the volumetric image data was deconvolved with *Huygens*. The deconvolved images were used during the manual correction process to simplify the visual detection of single cells in the captured microscope stacks. The algorithm parameters can be found in Tab. S6.

Dataset annotation The proposed dataset annotation was iterated three times. In the first iteration, no machine learning model was available. Instead, a subvolume of an imaged biofilm was segmented with *BiofilmQ*. The segmentation was manually reviewed and segmentation errors corrected with *napari*. Based on the resulting segmentation labels, a first *Stardist* model was trained. The trained model generates a prediction for the second iteration. To reduce the manual labelling and review effort, the proposed GUI identifies and classifies the emerging segmentation conflicts. Depending on the user input and the conflict topology, the conflict is either resolved automatically or marked for manual human intervention. After the manual correction of the remaining conflicts, the iteration is repeated until five different biofilms are fully annotated and can be used as training data for the different deep learning algorithms.

Data preparation To test the final segmentation accuracy on full microscope volumes, the biofilm volumes were split into two groups. Four biofilms were part of the model training group and were used for the training and validation of the deep learning models. One biofilm was hold back to evaluate the segmentation performance on a full biofilm which was not included in the training data. The training volumes were further split into smaller training patches with a spatial dimension of $64 \times 128 \times 128$ voxels ($z \times y \times x$) each. To provide suitable validation data during the training, the patches were split by a 1:10 ratio into the validation and training dataset.

Cellpose Since *Cellpose* requires 2D input data, each training patch was sliced in x-, y-, and z-direction prior to the training session to generate a large input parameter space. Only 2D training patches, which contained one or more annotated objects, remained in the dataset. All empty patches were removed from the training and validation data.

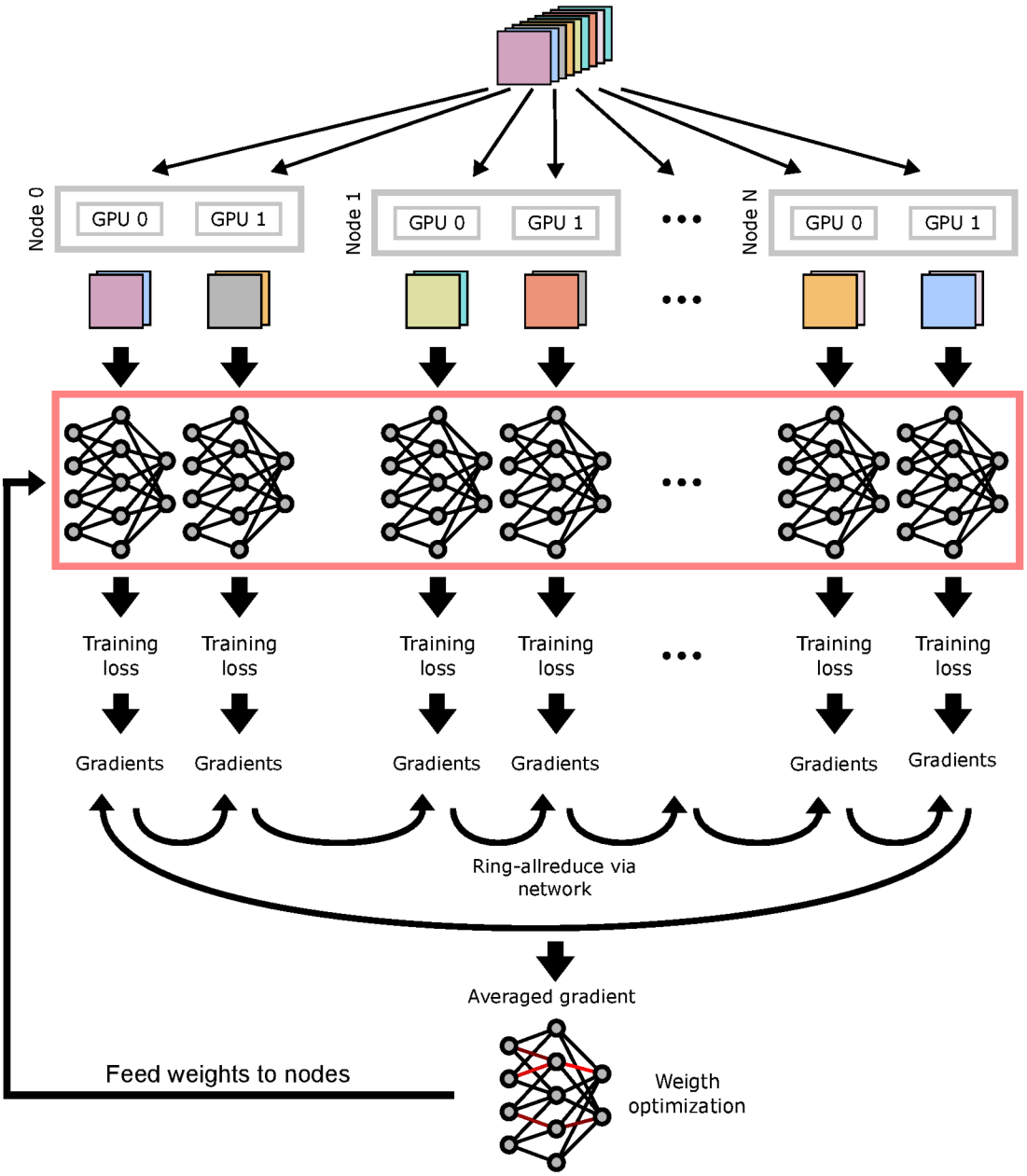


Figure 22: Schematic description of training with data parallelism on multiple GPUs. For each GPU a separate training process is started. Each training process loads a separate chunk of the overall training data. Each process calculates the training loss and the corresponding gradients individually. After each GPU has processed a mini-batch, the gradients are pooled together via a ring-allreduce function over the network. A single network is optimized with the calculated and averaged gradients. After the optimization, the new weights are sent to all GPU processes and the next training step is started.

To process all training patches in less than a week, a modification of the original *Cellpose* training process was required. The original *Cellpose* framework uses only a single GPU to calculate the network predictions, the training losses, and the optimization gradients. To speed up the neural network training, multiple GPUs can be used in parallel [93, 58]. One paradigm to achieve parallel processing of a large dataset is called data parallelism [93] and depicted in Fig. 22. The original dataset is split into smaller chunks. Each chunk is processed by a different worker. The workers are directly tied to a single GPU. The GPUs can either be installed in the same machine, or distributed on multiple processing nodes in a low latency network. After the prediction of the input data, each worker calculates the training loss and gradients of the network weights individually. In a next step, the gradients are gathered by a ring-allreduce approach and combined into a single gradient [131]. The averaged gradient is used to optimize the weights of a single network. After the optimization, the weights of all workers are set to the new optimized weights. This enables almost linear scaling of the training runtime with respect to the number of available GPUs [146].

Eight cluster nodes equipped with a total of 16 NVIDIA Quadro RTX5000 were used for the *Cellpose* training process. The shorter runtime enabled a hyper-parameter scan on a representative subset of the full training data. With the described approach, the optimal hyper-parameters for the custom *Cellpose* implementation were determined. To rule out possible negative side effects of the custom multi-GPU implementation, the original *Cellpose* training was iteratively restarted until the full dataset was processed. This enabled the direct comparison of the single- and multi-GPU training sessions.

Stardist In contrast to *Cellpose*, *Stardist* requires 3D training patches and does not normalize the input data by default. A percentile-based normalization has proven useful in previous studies [171]. In short, the percentile ranks for the lowest 1 % and the highest 99.8 % were calculated. All values above the higher rank were scaled to 1, all values below the lower rank were set to 0. All values between the ranks were scaled with respect to the new image range.

While *Cellpose* includes online data augmentations, *Stardist* requires user-defined augmentation functions [147]. Due to the observed symmetries during biofilm growth and imaging, random axis changes along the x- and y-axis and mirroring along the x- and y-axis were applied. To make the model more robust to the different SNRs along the z-axis, the intensity of the input image I was randomly rescaled according to

$$I' = I \cdot f_{\text{noise}} + s_{\text{noise}}, \tag{28}$$

with the global intensity factor $f_{\text{noise}} \in [0.6, 2.0)$ and a global offset $s_{\text{noise}} \in [-0.2, 0.2)$. The axis swaps and mirror operations were 1:1 replicated on the label image.

Prior to the training of the *Stardist* model, the ray number hyper-parameter has to be set according to the expected object shapes. For this purpose, all objects in a random selection of the validation data were modelled with star-convex shapes

Table 5: Key figures for different iterations of the iterative annotation approach.

Iteration	1	2	3
Prediction Model	BiofilmQ	Stardist	Stardist
Annotated biofilms	< 1	1	5
Predicted objects	3.2×10^3	10×10^3	25×10^3
Conflicts	-	5×10^3	10×10^3
Automatic split	-	1.8×10^3	2.2×10^3
Automatic merge	-	2.6×10^3	6.2×10^3
Manually corrected objects	3.2×10^3	926	1.5×10^3
Resulting training patches	7	320	640
Patch size (z, y, x)	100, 476, 102	64, 128, 128	64, 128, 128
Individual labels	2.9×10^3	7×10^3	19×10^3

and the mean IoU value between annotated and modelled object calculated. The model accuracy increases with the ray number. A disadvantage of a large ray number is a longer runtime and a larger GPU memory footprint. An optimal trait-off between model accuracy and hardware restrictions is given by a value just before the mean IoU values reaches a plateau and still fits into the available GPU memory.

BiofilmQ To compare the resulting accuracies of the *Stardist* and *Cellpose* models with our current segmentation pipeline, the in Sec. 4 presented *BiofilmQ* single-cell segmentation was used. To provide the best segmentation results, the parameters were manually tuned.

Scaling with training data abundance The best performing model was retrained with different fractions of the available training data to test the achievable accuracies based on the number of annotated cells.

5.5 Results

Iterative pipeline reduces manual annotations

The key results for each iteration cycle of the data annotation pipeline are summarized in Tab. 5. In the first iteration, *BiofilmQ* segmented approximately 3.200 objects in the selected biofilm sub-volume. After the manual review and correction in *napari*, approximately 2.900 objects remained in the dataset.

Based on the (semi-) manually annotated volume, seven training patches with the spatial dimensions $100\,\text{px} \times 476\,\text{px} \times 102\,\text{px}$ ($z \times y \times x$) could be extracted. Five patches were used for the training of a *Stardist* model, a single patch was used for the model validation, and the remaining patch was used for testing the segmentation accuracy. After the training process, the trained model was applied

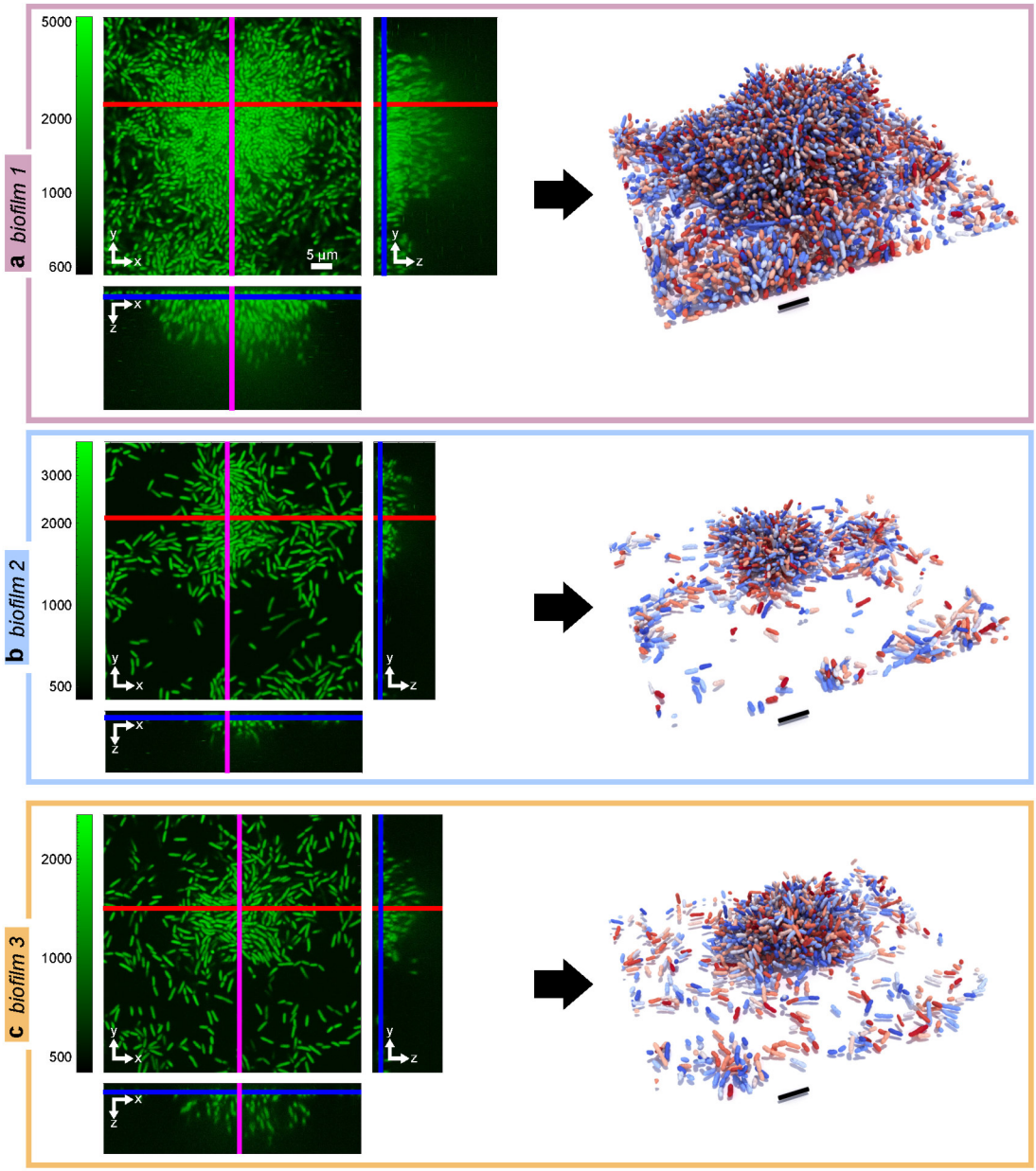


Figure 23: Visualizations of microscope stacks and rendered 3D annotations *biofilm 1 - 3*. The annotations were the result of the third iteration of the proposed iterative data annotation approach. Scale bar: 5 µm.

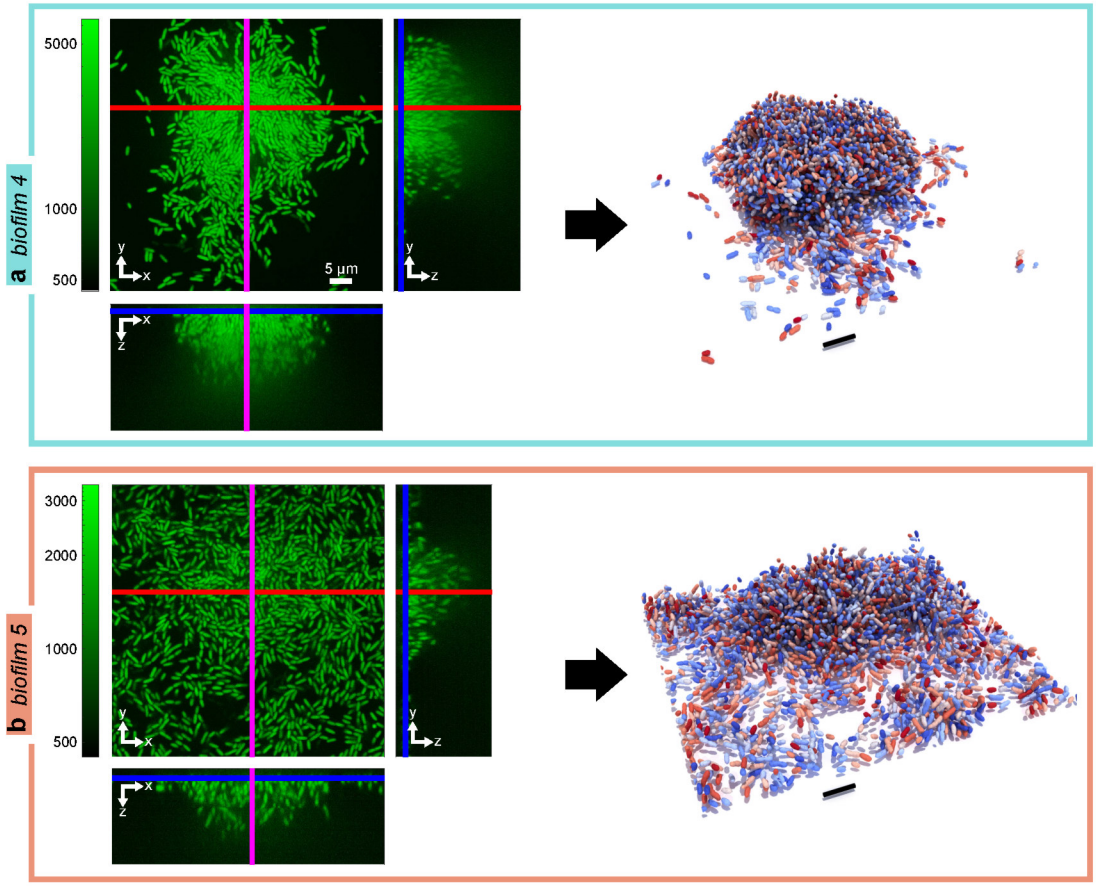


Figure 24: Visualizations of microscope stacks and rendered 3D annotations *biofilm 4* and *biofilm 5*. The annotations were the result of the third iteration of the proposed iterative data annotation approach. Scale bar: 5 µm.

on the full biofilm volume which was used in the first iteration. In the full volume, *Stardist* predicted $\sim$10.000 individual objects. The automatic error detection pipeline identified $\sim$5.000 conflicting segmentations. With the implemented GUI (Fig. 21c), $\sim$1.800 conflicts could be solved with an automatic split operation and $\sim$2.600 conflicts were resolved with an automatic merge operation. Only 926 conflicts needed manual correction in *napari*. After the second iteration and the manual correction, $\sim$7.000 different objects remained in the dataset. The annotation time linearly scales with the number of manual interventions. While keeping the number of manually corrected objects constant, the biofilm volume increased by a factor of five. Thus, I estimate that the annotation pipeline resulted in $\sim$ 5$\times$ speed-up.

The single annotated biofilm was split into 320 training patches with a spatial dimension of $64 \times 128 \times 128$ ($z \times y \times x$), each. A second *Stardist* model was trained on the patches and a third annotation iteration was started. The model was used to predict unannotated biofilm stacks. In the five predicted biofilms, $\sim 25 \times 10^3$ single object could be detected. The automatic error prediction pipeline detected $\sim$10.000 conflicts. With the GUI interface, 8.400 conflicts could be resolved automatically. Only $\sim$1.500 objects needed manual correction.

With the iterative pipeline $\sim$19.000 single cells could be annotated. A skilled annotator can process $\sim 30 - 40$ objects per hour. With the GUI interface, in contrast, $400 - 500$ conflicts per hour can be categorized. Since most conflicts can be resolved automatically, only a small fraction requires manual inspection. To replicate the results of the third iteration from scratch, three months of full-time annotation would be required. Instead, it took 1.5 weeks to review the remaining conflicts. This example illustrates the potential benefit of the semi-manual segmentation approach.

The five resulting (semi)-manually annotated biofilms are depicted in Figs. 23 & 24. In the following paragraphs, these biofilms will be used as training and test data for deep-learning-based segmentation pipelines.

Data parallelism reduces *Cellpose* runtime

The combination of three optimizer hyper-parameters namely learning rate, weight decay, and momentum were tested with respect to the measured mean average precision [172]. For each parameter combination, three networks were trained on a randomly selected subset of 5 % of the training data described above. The learning rate was varied in five logarithmic steps between 0.05 and 0.003125 (default value according to *Cellpose* publication: 0.025 [156]). Here, the learning rate is normalized with respect to the batch size [112]. Values between 10^{-6} and 10^{-1} were tested for the optimal weight decay (default: 10^{-5}). The momentum was tested for an optimal value as well. The momentum parameter space consisted of five values between 0.4 and 0.9. The full parameter scan resulted in 375 trained networks. The mean accuracy was tested on *biofilm 1* (Fig. 23a) which was not part of the training data.

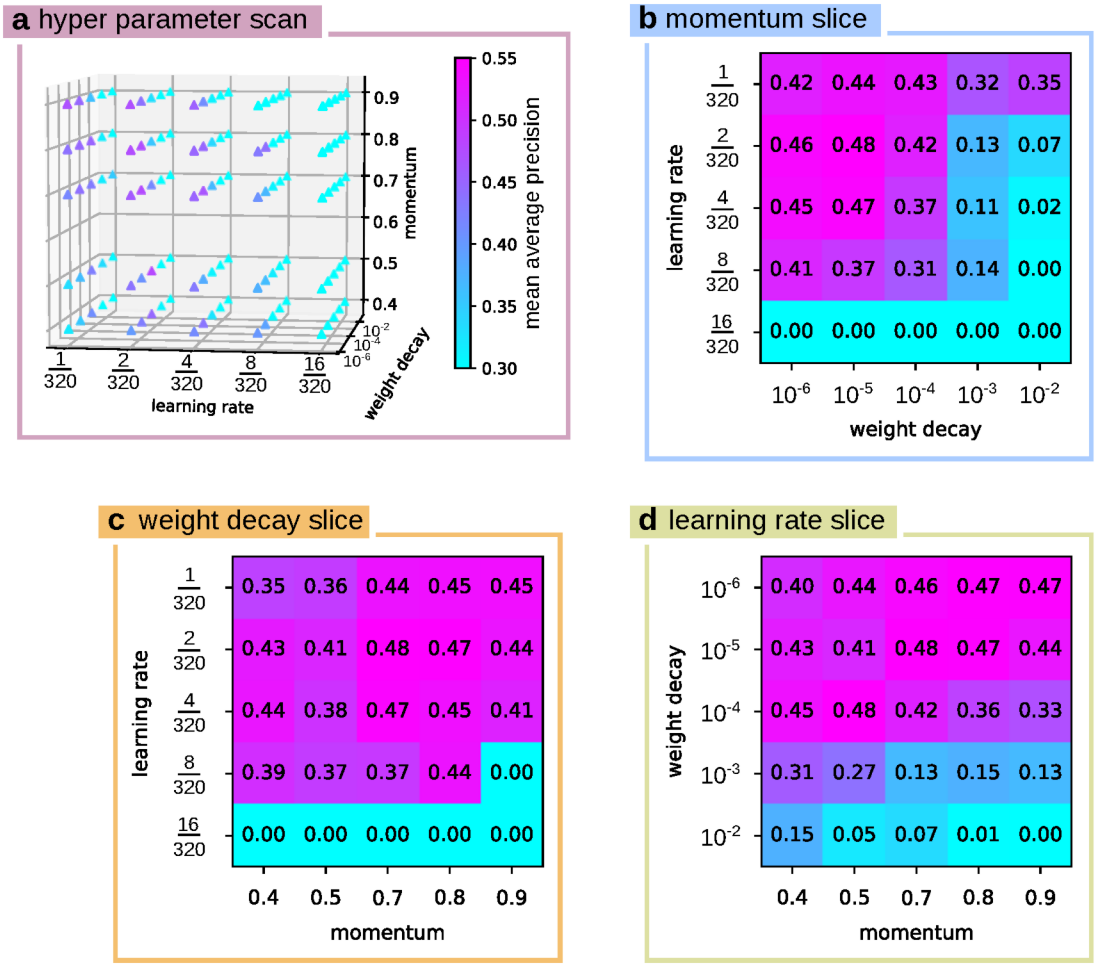


Figure 25: Hyper-parameter scan results for multi-GPU *Cellpose* training. Each value represents the mean of three different average precision calculations of a separate hyper-parameter combination. **a** Full parameter space of the three different hyper-parameters. Each marker represents a separate parameter combination. The mean average precision is colour-coded. The highest mean value could be found at a learning rate of 2/320, a weight decay of 10^{-5}, and a momentum of 0.5. **b** 2D representation of mean accuracy values in the learning rate/weight decay plane with a fixed momentum of 0.5. **c** Representation of the mean accuracy values at a fixed weight decay of 10^{-5}. **d** Mean accuracy values at a fixed learning rate of 2/320 for different weight decay and momentum combinations.

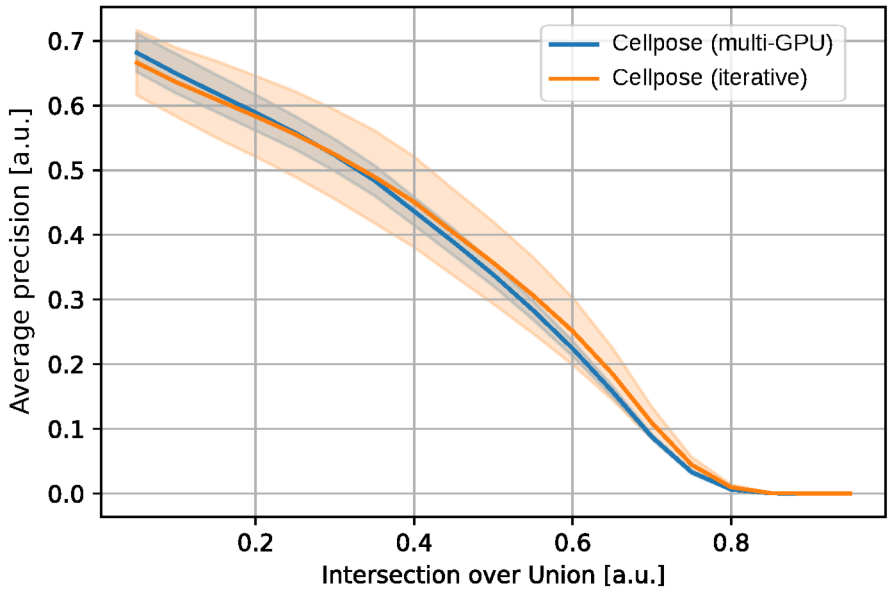


Figure 26: Comparison of the mean accuracy for the single-GPU (iterative) (n = 3) and multi-GPU (n = 5) *Cellpose* implementation on the 100 % of the available training data for different intersection over union thresholds. The transparent area indicates the standard deviation of the different replicates. The opaque line indicates the mean value over all replicates.

The resulting mean values for the three independent networks in the 3D parameter space are shown in Fig. 25. Based on the scan results, the highest mean average precision value could be found with a learning rate of 0.00625, a weight decay of 10^{-4} and a momentum of 0.5. All parameters differ from the *Cellpose* default parameter.

The orthogonal values around the optimal hyper-parameter combination are shown in three slices in Fig. 25b-d. For a simple visualization, the values are rounded to two decimal positions. Fig. 25b shows the values in the learning rate/ weight decay plane with a fixed momentum of 0.5. Fig. 25c shows the learning rate/ momentum space with a fixed weight decay of 10^{-4} and Fig. 25d shows the weight decay/ momentum space with a fixed learning rate of 0.00625.

To investigate the performance differences between the customised multi-GPU the original *Cellpose* implementation, an iterative start of the original *Cellpose* implementation with default parameters was conducted. The training of 50 consecutive epochs on the full dataset needed $\sim$21 GPU hours. The runtime limit for a single cluster job required a training break after 24 h. *Cellpose* requires 500 epochs for a full training session [156]. Therefore, after training a single network for 50 epochs, the weights had to be saved on disk, and reloaded for the next 50 training epochs. This approach is repeated until 500 epochs are reached. Due to the learning rate warm-up and cool-down, the iterative training differs slightly from the all-in-one training on 500 epochs [156, 58]. Depending on the overall cluster utilization, this iterative training approach takes 10-14 days for a single replicate. For the presented investigation, three replicates were trained in parallel.

The multi-GPU implementation uses 16 GPUs at once. As predicted, the new implementation completed the 500 training epochs within the runtime limit. For the purpose of this investigation, five replicates were trained on the previously described cluster hardware. The accuracy comparison of the two training approaches are shown in Fig. 26. The results indicate that the different training approaches produced equivalent mean average precision within the margin of error. While the iterative approach seems to produce slightly higher accuracy measurements, the multi-GPU implementation is more stable (smaller standard deviation) and thus provides more robust results with respect to single training sessions.

The results indicate the equivalence of the two training paradigms with respect to the achievable segmentation accuracy. The multi-GPU implementation, however, comes with the additional benefit of a huge decrease in the needed training time. This training time reduction comes with additional requirements on the used compute infrastructure. Regular workstations are equipped with up to four GPUs. The used set up additionally requires low latency network devices such that only GPU clusters or cloud instances can be used. This adds another layer of complexity to the training of the presented *Cellpose*-based segmentation.

Stardist **exceeds** ***BiofilmQ*** **segmentation accuracy**

The results of the *Stardist* ray number estimation are shown in Fig. 27. The ground truth data were replicated once with the assumption that the voxel are spaced isotropically in the image stack (i.e. all directions are recorded with the same voxel distance) and once with the correct anisotropic voxel spacing (61 nm in x- and y-direction and 100 nm in z-direction). The results indicate that an anisotropic distribution of the ray directions results in slightly better results than the isotropic one. The measurement indicates further, that only slight improvements in the mean IoU are expected if more than 192 rays are used. Therefore, all following *Stardist* networks were trained with 192 different ray direction. On the presented ground truth data, the mean IoU is restricted to a mean value of ~ 0.81.

Similar to the *Cellpose* segmentation approach, five *Stardist* replicates were trained on the available training data. While the training data was split into 64 px $\times$ 128 px $\times$ 128 px ($z \times y \times x$) patches, the network used a random crop of the dimensions 48 px $\times$ 96 $\times$ 96 px ($z \times y \times x$) to add additional variation to the training data. All networks were trained with a batch size of two and the default *Stardist* hyper-parameter since the no architectural changes were made. The networks were trained for 400 epochs with 100 steps each.

A direct comparison of the achievable segmentation accuracies with the different segmentation pipelines can be found in Fig. 28. The results indicate that the *Cellpose* model cannot segment non-isotropic biofilm data as well as the other two contestants. In the original *Cellpose* publication [156], no differences between the different 2D slices due to anisotropic effects are shown. The used biofilm data,

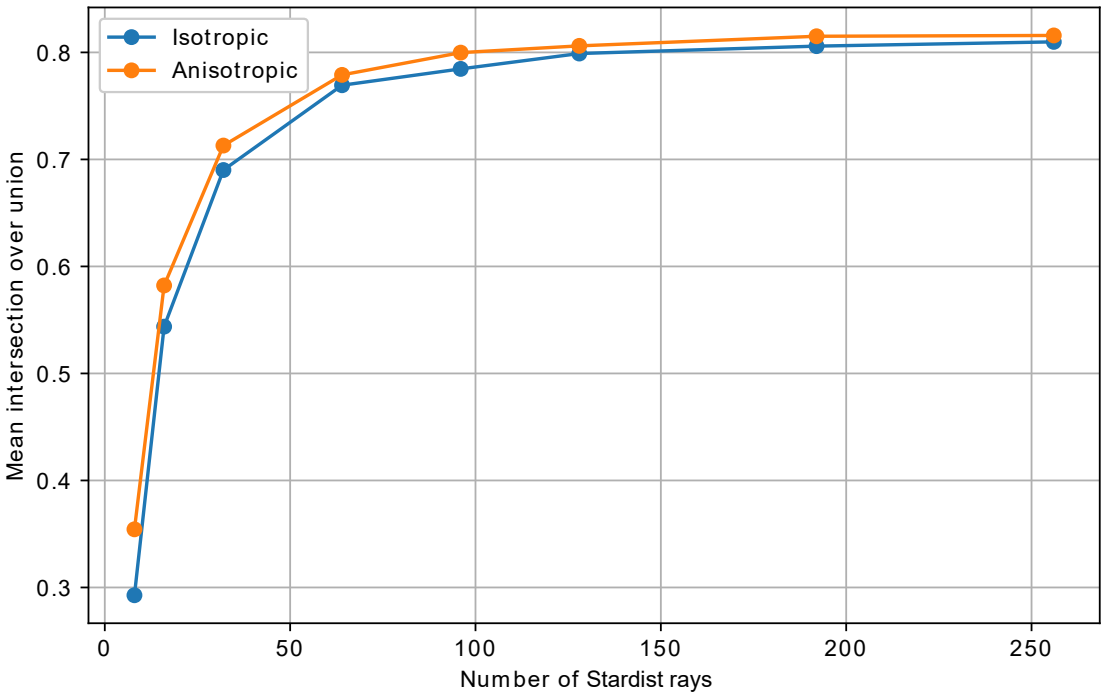


Figure 27: Model accuracy calculated for different *Stardist* ray numbers and ray distributions. Based on the validation data, a representative dataset was selected and remodelled with star-convex polygons with different corner directions and corner distributions. The intersection over union was calculated and averaged for all objects in the dataset. The isotropic model distributes the corners evenly over all spatial directions according to a spherical Fibonacci lattice. The anisotropic model takes the different voxel spacing in x- and y-, and z-direction into account.

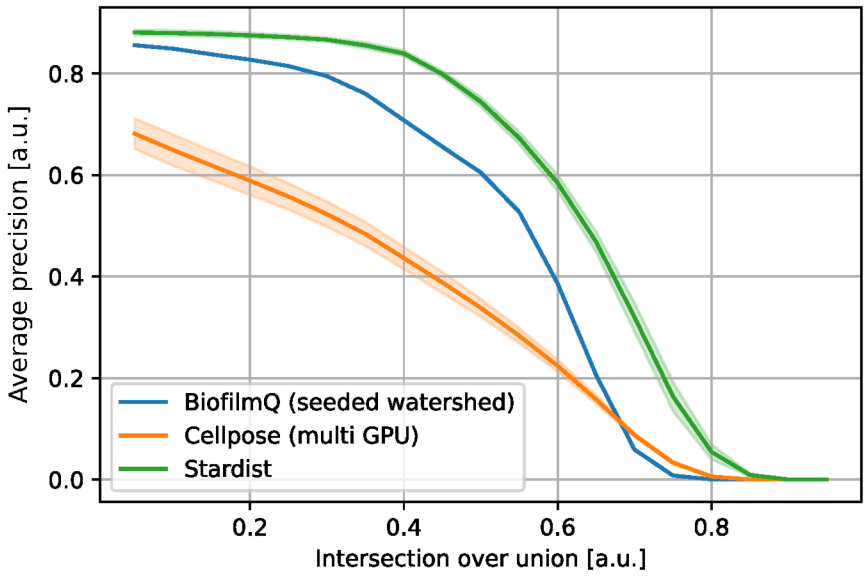


Figure 28: Average precision calculation for different IoU thresholds. The *BiofilmQ* segmentation n = 1) was performed with a manually fine-tune segmentation parameter set and used the latest instance segmentation approach. The *Cellpose* segmentations ($n = 5$) were trained on multiple GPUs to shorten the training time on the full dataset. The *Stardist* models ($n = 5$) used 192 rays and the same dataset as the *Cellpose* implementation for the model optimizations.

in contrast, is highly anisotropic. Due to the large PSF-based distortions along the z-direction, images in the xy-plane notably differ from images in the xz- and yz-plane. It can be assumed that even with a large amount of data, these effects severely influence the segmentation performance of *Cellpose* in 3D.

In contrast to *Cellpose*, the *Stardist* segmentation produces superior results to the *BiofilmQ* pipeline with the proposed seeded-watershed extension. All five replicates produce similar results. Stochastic effects during the network training affected the training outcome only slightly. Without further optimizations by hyper-parameter tuning, *Stardist* out-performed the carefully programmed classical segmentation pipeline. This example illustrates the benefit and versatility in 3D segmentation pipelines with deep learning techniques.

To make the difference in segmentation qualitatively visible, Fig. 29 illustrates the FN and FP detections of the *BiofilmQ* segmentation (Fig. 29a & b) and one representative *Stardist* segmentation (Fig. 29c & d) at an IoU threshold of 0.5. All renderings show segmentations of *biofilm 1*. While the representation of the FN detection is based on the ground truth volume (Fig. 29a & c), the FP visualization shows the segmentation results of the stated segmentation pipeline. All visualizations contain a partially-cut biofilm, such that not only the cells at the outside but also the inside of the biofilms can be investigated for detection errors.

The FN visualization of the *BiofilmQ* segmentation pipeline (Fig. 29a) shows that a large number of cells at the top of the biofilm could not be detected

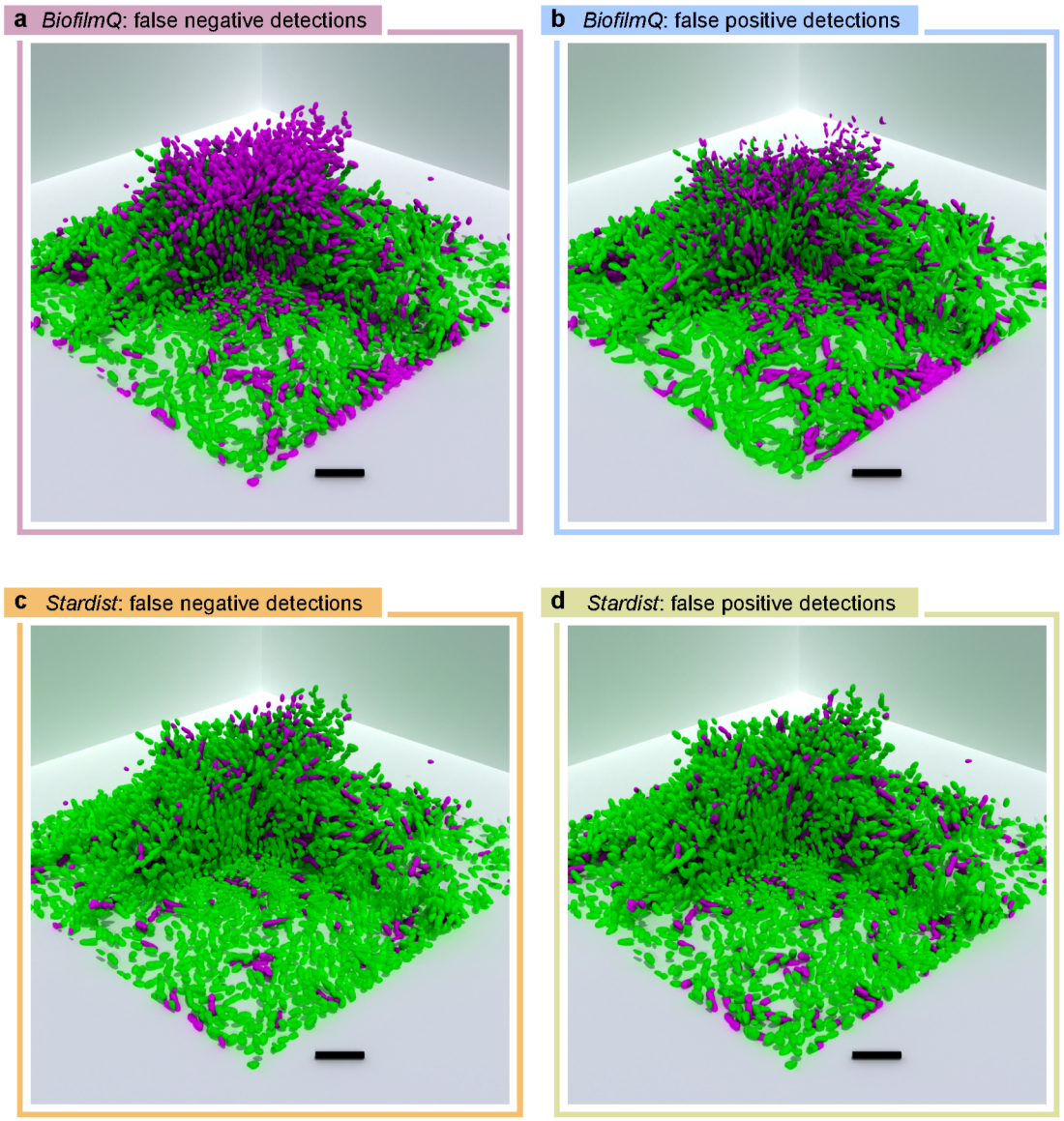


Figure 29: Rendered visualization of *BiofilmQ* and *Stardist* segmentation errors. One quadrant of the segmentations is removed for better visualization of errors inside the biofilm volume. **a** (Magenta) False negative (FN) detections in the *BiofilmQ* segmentation. **b** False positive (FP) detections (Magenta) in the *BiofilmQ* segmentation. **c** FN detections for the *Stardist* segmentation (Magenta). **d** FP detections of the *Stardist* segmentation pipeline. In all panels, the true positive (TP) detections are shown in green. Scale bar: 5µm.

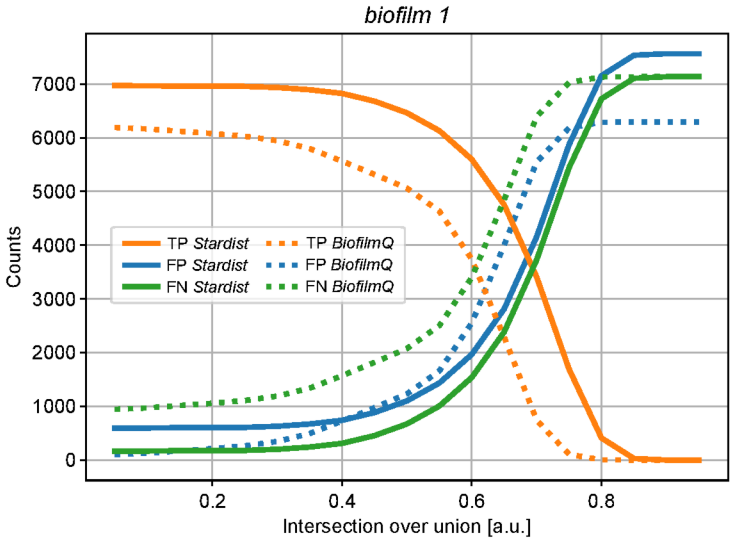


Figure 30: Resulting true positive (TP), false positive (FP), and false positive (FP) object detections in the *BiofilmQ* and *Stardist* segmentations of *biofilm 1* in dependence on the intersection over union (IoU) threshold. Only the best replicate is shown for each segmentation pipeline.

correctly. This is probably due to the reduced SNR in the higher layers of the microscope volume and the previously discussed global background threshold (Sec. 4.2). Besides problems with the segmentation of high biofilms, *BiofilmQ* shows segmentation errors in the core of the biofilm and the bottom layer. At least in the biofilm slices close to the substrate, a high SNR is expected. These miss-segmentations can be attributed to under-segmented cell clusters and are investigated in Fig. 31.

Based on *biofilm1* as input data, *BiofilmQ* also produces a large quantity of FP detections (Fig. 29b). The locations of the FP errors closely match the ones of the FN errors. This indicates that the *BiofilmQ* pipeline is able to detect the presence of cells but does not produce a high IoU with the ground truth data. While the *BiofilmQ* segmentation closely match the overall biofilm topology, cells at the top of the biofilm are either too small or omitted at all.

In contrast to the *BiofilmQ* segmentation results, the *Stardist* segmentation produces less FN detections (Fig. 29c). Almost all cells in the biofilm core are segmented correctly, and the *Stardist* segmentation only misses some cells at the top of the biofilm. In the FP detections (Fig. 29c) the same pattern repeats: *Stardist* manages to segment a large fraction of cells and the FP and FN locations are closely correlated.

Fig. 30 quantitatively supports the qualitative observations. Here, the numbers of TP, FP, and FN detections are plotted for both the segmentation pipelines above the different IoU thresholds. For all investigated IoU values, the trained *Stardist* networks produce a higher number of TP detections than *BiofilmQ*. The number of objects in the ground truth dataset accounts to 7.135. At the reference IoU value of 0.5 *Stardist* detects $\sim$90.6% (6.463 objects) of all ground truth objects. In contrast, *BiofilmQ* detects only 70.9% (5.060) of the objects at the

same IoU threshold correctly. The FP rate of both segmentation approaches are comparable. *Stardist* detected 1.101 additional objects (equals to 15.4%) and *BiofilmQ* 1.231 (17.3%). The ratio of not detected ground truth objects (FN) was measured with 9.4% (672 cases) for *Stardist* and 29.1% (2.075 cases) for *BiofilmQ*.

The results support the qualitative impression, that *Stardist* produces superior segmentation results in comparison to *BiofilmQ* and that the deep learning pipeline is more robust on the wide range of SNRs occurring in the biofilm volume. The results in Fig. 30 also indicate that *Stardist* tends to over-segment the test data. While *BiofilmQ* produces an under-segmented object detection. To further qualitatively compare the errors an investigation on a per-object base is necessary.

Examples of randomly selected errors are shown in Fig. 31. While the selection was randomized, the cases were re-arranged to illustrate the similarities. Each panel shows a separate pipeline and error class combination. For each combination, three separate errors were selected and the xy-, xz-, and yz-plane at the centroid of the marked object is displayed. The ground truth is indicated with a blue, the segmentation result with a red outline. The FN errors for *BiofilmQ* include IoU values of $0.46 - 0.27$ (Fig. 31). The first example shows an object which was correctly detected, but whose outline does not match the outline of the ground truth annotation. The second case shows the *BiofilmQ* detection of an under-segmented single object. Instead of a single object, the ground truth annotation contains two separate ones. In the third case, *BiofilmQ* over-segmented an object into two. In this case, the ground truth annotation only contains a single object.

In the randomly selected FP cases of the *BiofilmQ* segmentation, similar error classes can be observed. The first case shows a misshaped correct detection of a single cell in the ground truth data. The second case shows an under-segmented object and in the third case a misshaped object due to the under-segmentation of a neighboring cell is visible. Note that beside the selected errors other erroneous segmentation are visible in the example sets, but not included in the given IoU measurement.

The *Stardist* segmentation exhibits only two error types: Misshaped segmentations, and over-segmentations. The first example of the FN detection is a misshaped segmentation (Fig. 31c). While an object was correctly placed at approximately the same position, the exact shape of the ground truth data is not matched exactly. The remaining two example cases contain over-segmented cells. This is the most common segmentation error in *Stardist* segmentations. Instead of predicting one star-convex shape, *Stardist* detects two separate objects in the same ground truth cell. The same error class can be observed in the FP examples. Every single example object is over-segmented which results in additional objects in the segmentation results. The additional object do not exhibit a large overlap with any ground truth object and are thus counted as FP.

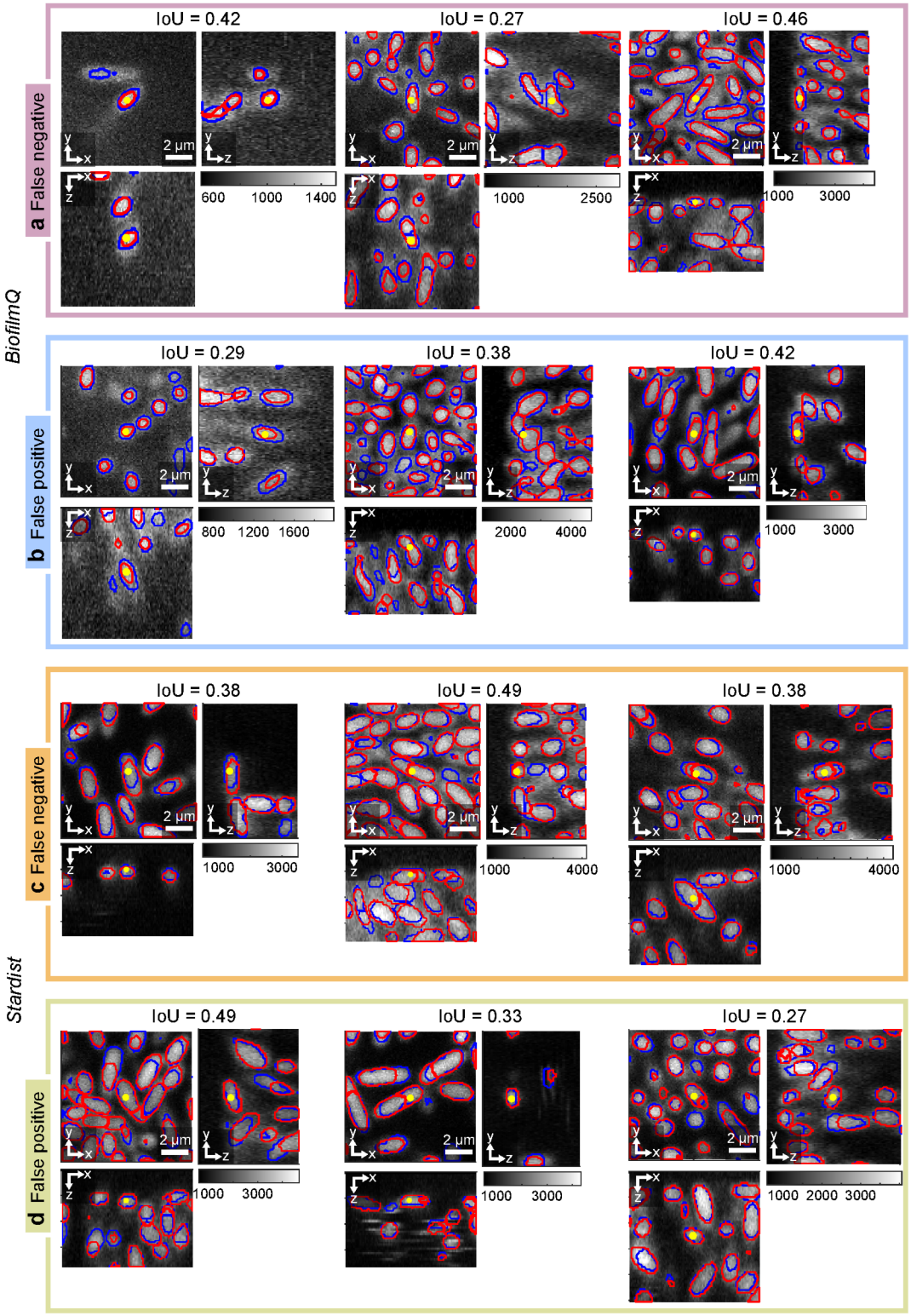


Figure 31: Single-cell segmentation error examples for *BiofilmQ* (**a** & **b**) and *Stardist* (**c** & **d**) segmentations. For the different error types false negative (FN) and false positive (FP) three randomly selected examples are shown (selected objects are marked with yellow dot). The xy-, xz-, and yz-plane through the marker are given for the 3D representation of the error. The segmentation result is shown as red outline; the ground truth annotation as blue outline. The raw microscope data was rescaled for maximum contrast and interpolated in z for isotropic resolution.

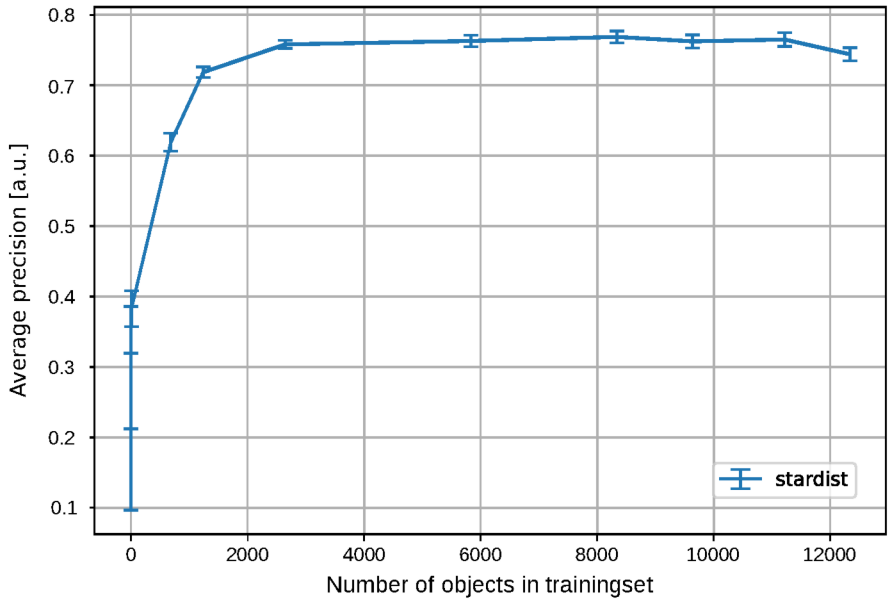


Figure 32: Influence of the number of annotated objects in the training data on the *Stardist* segmentation accuracy. Five different network instances were trained for each object count. The accuracy was measured at an intersection over union (IoU) threshold of 0.5. Test dataset was biofilm 1.

Stardist **achieves robust segmentations with 3000 annotated cells**

Based on the generated training data and the superior results of the *Stardist* segmentation pipeline, the question on the robustness of the segmentation in relation to the number of annotated objects arises. To measure the *Stardist* performance, the neural network was trained from scratch with different fractions of the original training data. For each fraction, five replicates were trained to investigate the robustness in different training sessions.

Fig. 32 shows the dependence of the mean accuracy score reached by the trained *Stardist* networks in dependence on the number of individual objects in the corresponding training data. After a steep increase for a small number of training patches, the mean accuracy reaches a plateau when trained with $\sim$3000 single cells in the training data. Even with more training data, the segmentation accuracy cannot be improved further. In the case of the investigated biofilm $\sim$3000 objects are sufficient to reach the observed accuracy above the currently existing classical segmentation pipelines. This means that technically a single medium-sized biofilm contains enough objects to reach high segmentation accuracies. In practice, however, it is probably more beneficial to include parts of different biofilms in the training set to increase the variance in the training data and make the resulting segmentation more robust to variations in the experimental set up.

5.6 Conclusion

In the presented chapter, I introduced an iterative annotation approach to quickly annotate a large number of single objects in 3D biofilm data. By using deep learning segmentation and automatic error detections during the data annotation, the required annotation time for generating a suitable large training dataset could be reduced by a factor of five. The method is easy to use and enables even a small research group with limited resources to create a training dataset for high accuracy segmentation networks.

The annotation pipeline enabled the generation of high-quality training data with $\approx 19 \times 10^3$ single-cell objects in five dense microbial communities. The data can be used for validating existing segmentation pipelines or for the training of proposed deep learning segmentations from scratch. The importance of high quality annotated datasets will increase in all future single-cell segmentation projects. By accumulating a large number of annotated objects, new pipelines can quickly be evaluated and new machine pipelines will be more robust for a large variety of imaging conditions.

Based on the generated training data, I trained a custom build multi-GPU implementation of the recently announced *Cellpose* architecture. The new implementation enabled the training with data-parallelism on several GPUs and reduced the training time by a factor of ten. The reduced training time enabled a hyper-parameter scan on a representative fraction of the full dataset. The hyper-parameter scan revealed different parameter choices than previously noted in literature [156]. By training on the full training data, I could verify that the multi-GPU implementation exhibits the same segmentation performance as the default *Cellpose* implementation. Thus, the reduced training time results in faster iteration cycles for exploring the limitations of the architecture or fast incorporation of new training data if necessary.

The average precision of *Cellpose* segmentation cannot compete with the ones measured for *BiofilmQ* or *Stardist*. One reason could be the use of a single 2D prediction network along all spatial dimensions. When closely investigated along the different imaging axes, a clear difference between the different dimensions for the used biofilm data becomes apparent. At the time of the experiments, no anisotropy option was included in the *Cellpose* library. However, the authors recently added such a feature in the current release[11].

Even with an included anisotropy correction, it remains unclear whether the observed average precision differences will disappear. The PSF-related distortions along the z-axis still break the overall symmetry in the image data. In my opinion, a combination of two *Cellpose* networks could potentially result in a high performance improvement. Instead of training a single network on the xy-, xz-, and yz-plane, one network is exclusively trained on lateral and a second network exclusively trained on the axial image planes. By exploiting the vector addition for the 3D segmentation generation, the results of both can be combined into a

[11] `https://github.com/MouseLand/cellpose/issues/106`

single segmentation without rewriting the entire *Cellpose* code base.

Similar to the *Cellpose* models, the *Stardist* networks were trained on the new annotated data. The accuracy measurements revealed that the *Stardist* architecture produces superior results in comparison to one of the best currently available approaches for dense microbial communities image segmentation [66]. In a direct comparison of the segmentation approaches, the experiments revealed, that the in Sec. 4.5 implemented pipeline is unable to reconstruct the volume of cells in higher layers of large biofilms. *Stardist*, however, correctly segments cells in the problematic layers and thus exhibits an improved overall segmentation accuracy for large biofilms.

The classical segmentation pipeline as well as the deep learning-based *Stardist* exhibits systematic segmentation errors. With the extension of the watershed instance segmentation (Sec. 4.5), *BiofilmQ* is prone to under-segmentation of single cells in the core of the biofilm community. While heuristics can partly correct these errors, under-segmentation is usually more difficult to mitigate than over-segmentation, especially if over- and under-segmentations are produced by the image analysis pipeline at the same time. *Stardist*, however, exclusively shows over-segmentation errors.

It can be hypothesized that the *Stardist* segmentation errors can be reduced further. One line of attack could be the automatic error detection (and correction) with a 3D classifier trained on a reasonably large set of joint image, label, and segmentation data. The classifier could predict probability that a given cell is over-segmented. This approach, however, would result in an additional post-processing step, which should be avoided to exploit the end-to-end application (and training) of the deep learning algorithms. Another approach to reduce the number of wrongly segmented cells might be connected with a modified loss function which increases the loss value for under-segmented cell. Additionally, the influence of the relatively small patch size of $48\,\mathrm{px}\times96\,\mathrm{px}\times96\,\mathrm{px}$ ($z \times y \times x$) on the segmentation results has not been investigated yet. While the patch size is well above the individual cell dimensions, edge effects at the boundaries of the training volumes could lead to an increased number of small cells. The patch size, however, is a technical limitation induced by the amount of available GPU memory and RAM of the compute machine. Larger patches would quickly exhaust the different memory types and require a complete revision of the training scripts and the fine-tuned prediction scripts.

The observed over-segmentations could also be explained with wrongly labelled training data. It is known that different subject analysis steps can introduce biases in biofilm research results [170]. This is even more important since small error in the training set can have severe implications in the application of deep learning algorithms [48, 22]. One way to improve the training data beyond the subject impression of a single annotator is the introduction of a majority voting scheme [157]. In this setting, multiple annotators correct themselves, which subsequently results in a more robust training dataset.

Finally, I could estimate that $\sim$3.000 annotated cells are sufficient to achieve

the stated segmentation accuracies. The saturation on the segmentation accuracy emerge due to the previously mentioned over-segmentation. Thus, accuracy improvements by using more training cannot be expected. Most likely, *Stardist* reached a capacity limit due to a limited number of free parameters [158] and a deeper backbone architecture should be tested. Another study for the *U-Net* architecture indicate that more training data can introduce additional over-segmentations to the segmentation results [22].

Overall, *Stardist* manages to detect the biomass of the predicted biofilm remarkably well. Therefore, I suggest a shift to deep learning algorithms for the segmentation of bacterial biofilms. Current classical segmentation pipelines will play a minor role in future high impact publications. Classical segmentation will remain the segmentation of choice for quick evaluations of experimental data when no annotated ground truth is available. Yet, the superior performance and generalization capabilities does not come without drawbacks. The annotation time, pipeline implementation, GPU set up, and interference requires much more time than the application of an already existing classical segmentation pipeline.

6 Summary and Outlook

The possibility to analyse confocal microscope images with single- or pseudo-cell resolution can provide answers on the spatio-temporal mechanisms inside bacterial biofilms. The mechanisms which can be discovered include the dependence of biofilm morphology on matrix production [41, 179] and nutrient availability [151]. Further it is important to investigate response to anti-microbial agents such as phages or antibiotics [164, 38]. With the accessibility of more and more single cell features, the full understanding of emerging phenotypes inside bacterial biofilms comes into reach [82].

A logical consequence is the description of the dynamics of the emerging phenotypes over time. To attribute these to the changing cell location relative to position inside the microbial community, the history of every single cells has to be known. Hence, the full lineage tree of the community is required. First steps into this direction have been made [134]. Yet, a complete tracking (i.e. such as available for eukaryotic systems [113]) is build on extensive engineering efforts [5, 6] and difficult to achieve in prokaryotic systems without powerful segmentation tools.

Although single cell segmentation algorithms have been used in the past, the adaptation to new experimental data required extensive image analysis knowledge. The creation of the presented *BiofilmQ* image analysis software democratizes the single-cell image analysis such that the algorithms become accessible to a wider user base. The GUI leads the user through the required steps for digital image cytometry such as file import, image preparation, segmentation, parameter calculation, object tracking, and visualization. This shortens the time between microscope experiment and analysis result significantly and enables the incremental improvement of experimental designs. Additionally, it provides batch capabilities, which (1) automatizes the analysis of large amount of experimental data and (2) can be easily reused for multiple experimental settings.

The evaluation of the current segmentation algorithms for biofilms revealed a huge tendency to over- and under-segment cells inside the biofilm core and failed to correctly detect cells in low SNR regions far away from the substrate. This work shows that the over-segmentation can be corrected by a careful selection of the watershed seeds. This improved the overall segmentation performance on the annotated test biofilms. The improvement becomes critical for the description of single-cell shapes inside microbial communities. To come once again a step closer to the trajectory description of single cells, an in-depth modification of the algorithm would be required such that a new paradigm for single cell segmentation inside bacterial biofilms became necessary.

Based on current best practices in biological single-cell segmentation [160], a deep learning-based approach was chosen. The end-to-end application of deep learning for single-cell segmentation on full sized bacterial biofilms could improve the segmentation accuracy beyond the level achievable by the mentioned classical algorithm extension. The 3D training data generation was made possible by an

iterative annotation approach with an automatic error detection. To provide a fair comparison between the two chosen deep learning segmentation approaches, a multi-GPU version of the popular *Cellpose* implementation [156] was created and a hyper-parameter scan conducted. The customized version produced the same segmentation accuracy as the default implementation, while providing linear runtime improvements with the number of used GPUs. However, the *Stardist* architecture [172] provided better segmentation accuracies on non-isotropic image data and even out-competed the already improved classical segmentation pipeline. An evaluation of the segmentation performance revealed that 3000 annotated cells would be sufficient to provide suitable training data for the network optimization. With the presented annotation pipeline, 3000 objects can be annotated in less than a week, given that a not-yet-optimized model exists. Based on the conducted analysis it can be concluded that deep learning algorithms can increase the accuracy of single-cell segmentation inside dense microbial communities and provide a possible path to full lineage tracking with SDCMs.

While the created training data was experiment specific, the used networks are probably capable to produce robust performance when trained on diverse training data captured under different imaging settings [22]. Although the deep learning field has seen huge changes in the last five years, the *U-Net* architecture [139] (and its variants) still dominate the biological image segmentation field. Present developments indicate the emergence of transformer based image predictions [27] and unsupervised image segmentation [175, 106]. These new approaches are rendered useless for the biofilm research community without training and benchmark datasets which contain actual biofilm image data.

Although deep learning models produce superior segmentation results, the time investment for the implementation can be substantially higher than the required time for fine-tuning an already existing classical segmentation pipeline [22]. It is often overlooked, that even with a small amount of training data, the segmentation parameters of a filter-based image analysis can be automatically tested. This is already possible with dataset sizes unsuitable for the optimization of a learning algorithm. An additional advantage is the much lower demand on the available compute resources when compared to a deep learning based hyper-parameter scan. There are some occasions, however, were the time (and compute) investment into deep learning segmentation algorithms are out-weighted by the higher segmentation accuracy. 1) The classical segmentation has to be created from scratch for a new experimental set up and 2) the additional segmentation accuracy is needed for the down-stream data analysis.

In the past, the fast improvements in computer vision were driven by the availability of large scale annotated training data [49, 103, 22]. A reasonably large set of densely annotated biofilm data could provide a reasonable opportunity for computer vision researchers to test and validate their algorithms in a challenging 3D setting. In return, the optimized and published algorithms benefit the biofilm research community with even more accurate results.

Bibliography

[1] Ernst Abbe. "Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung". In: *Arch. für mikroskopische Anat.* 9 (1873), pp. 413–468. ISSN: 0176-7364. DOI: 10.1007/BF02956173.

[2] David Acuna et al. "Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++". In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* (2018), pp. 859–868. ISSN: 10636919. DOI: 10.1109/CVPR.2018.00096. arXiv: 1803.09693.

[3] Aysun Adan et al. "Flow cytometry: basic principles and applications". In: *Crit. Rev. Biotechnol.* 37.2 (2017), pp. 163–176. ISSN: 15497801. DOI: 10.3109/07388551.2015.1128876.

[4] James Ahrens, Berk Geveci, and Charles Law. *ParaView: An End-User tool for Large Data Visualization*. Elsevier, 2005. ISBN: 978-0123875822.

[5] Fernando Amat et al. "Efficient processing and analysis of large-scale light-sheet microscopy data." In: *Nat. Protoc.* 10.11 (2015), pp. 1679–96. DOI: 10.1038/nprot.2015.111.

[6] Fernando Amat et al. "Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data". In: *Nat. Methods* 11.9 (2014), pp. 951–958. ISSN: 15487105. DOI: 10.1038/nmeth.3036.

[7] Ignacio Arganda-Carreras et al. "Trainable Weka Segmentation: A machine learning tool for microscopy pixel classification". In: *Bioinformatics* 33.15 (2017), pp. 2424–2426. ISSN: 14602059. DOI: 10.1093/bioinformatics/btx180.

[8] Min Bai and Raquel Urtasun. "Deep watershed transform for instance segmentation". In: *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017* 2017-Janua (2017), pp. 2858–2866. DOI: 10.1109/CVPR.2017.305. arXiv: 1611.08303.

[9] R. Bakke, R. Kommedal, and S. Kalvenes. "Quantification of biofilm accumulation by an optical approach". In: *J. Microbiol. Methods* 44.1 (2001), pp. 13–26. ISSN: 01677012. DOI: 10.1016/S0167-7012(00)00236-0.

[10] Athanasios D. Balomenos et al. "Image analysis driven single-cell analytics for systems microbiology". In: *BMC Syst. Biol.* 11.1 (2017), p. 43. ISSN: 1752-0509. DOI: 10.1186/s12918-017-0399-z.

[11] Francisco Balzarotti et al. "Nanometer resolution imaging and tracking of fluorescent molecules with minimal photon fluxes". In: *Science (80-.).* 355.February (2017), pp. 606–612. DOI: 10.1126/science.aak9913.

[12] Nick Barnes. "Publish your computer code: It is good enough". In: *Nature* 467.7317 (2010), p. 753. ISSN: 00280836. DOI: 10.1038/467753a.

[13] Christiane Barranguet et al. "Studying undisturbed autotrophic biofilms: still a technical challenge". In: *Aquat. Microb. Ecol.* 34 (2004), pp. 1–9. DOI: 10.3354/ame034001.

[14] Thomas M. Bartlett et al. “Article A Periplasmic Polymer Curves Vibrio cholerae and Promotes Pathogenesis Article A Periplasmic Polymer Curves Vibrio cholerae and Promotes Pathogenesis”. In: *Cell* 168.1-2 (), 172–185.e15. ISSN: 0092-8674. DOI: 10.1016/j.cell.2016.12.019.

[15] Saima Ben Hadj et al. “Blind restoration of confocal microscopy images in presence of a depth-variant blur and Poisson noise”. In: *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.* (2013), pp. 915–919. ISSN: 15206149. DOI: 10.1109/ICASSP.2013.6637782.

[16] Stuart Berg et al. “Ilastik: Interactive Machine Learning for (Bio)Image Analysis”. In: *Nat. Methods* 16.12 (2019), pp. 1226–1232. ISSN: 15487105. DOI: 10.1038/s41592-019-0582-9.

[17] Olga Besharova et al. “Diversification of gene expression during formation of static submerged biofilms by Escherichia coli”. In: *Front. Microbiol.* 7.OCT (2016), pp. 1–17. ISSN: 1664302X. DOI: 10.3389/fmicb.2016.01568.

[18] Sinem Beyhan and Fitnat H. Yildiz. “Smooth to rugose phase variation in Vibrio cholerae can be mediated by a single nucleotide change that targets c-di-GMP signalling pathway”. In: 63.January (2007), pp. 995–1007. DOI: 10.1111/j.1365-2958.2006.05568.x.

[19] Steve Branson, Pietro Perona, and S. Belongie. “Strong supervision from weak annotation: Interactive training of deformable part models”. In: *Proc. IEEE Int. Conf. Comput. Vis.* (2011), pp. 1832–1839. DOI: 10.1109/ICCV.2011.6126450.

[20] Tim-Oliver Buchholz et al. “DenoiSeg: Joint Denoising and Segmentation”. In: Dl (2020). arXiv: 2005.02987.

[21] Juan C. Caicedo et al. “Data-analysis strategies for image-based cell profiling”. In: *Nat. Methods* 14.9 (2017), pp. 849–863. ISSN: 15487105. DOI: 10.1038/nmeth.4397.

[22] Juan C. Caicedo et al. “Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images”. In: *Cytom. Part A* 95.9 (2019), pp. 952–965. ISSN: 15524930. DOI: 10.1002/cyto.a.23863.

[23] Juan C. Caicedo et al. “Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl”. In: *Nat. Methods* 16.12 (2019), pp. 1247–1253. ISSN: 15487105. DOI: 10.1038/s41592-019-0612-7.

[24] Anne E. Carpenter, Lee Kamentsky, and Kevin W. Eliceiri. “A call for bioimaging software usability”. In: *Nat. Methods* 9.7 (2012), pp. 666–670. ISSN: 15487091. DOI: 10.1038/nmeth.2073.

[25] Bi Chang Chen et al. “Lattice light-sheet microscopy: Imaging molecules to embryos at high spatiotemporal resolution”. In: *Science (80-.).* 346.6208 (2014). ISSN: 10959203. DOI: 10.1126/science.1257998.

[26] Jianxu Chen et al. “The Allen Cell Structure Segmenter: a new open source toolkit for segmenting 3D intracellular structures in fluorescence microscopy images”. In: *bioRxiv Prepr.* (2018). DOI: 10.1101/491035.

[27] Mark Chen et al. “Generative Pretraining From Pixels”. In: *ICML*. 2020.

[28] Özgün Çiçek et al. “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”. In: *Med. Image Comput. Comput. Interv. – MICCAI 2016*. Ed. by Sebastien Ourselin et al. Springer International Publishing, 2016, pp. 424–432. ISBN: 9783319467238. DOI: 10.1007/978-3-319-46723-8.

[29] Dan C. Ciresan et al. “Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images”. In: *Adv. Neural Inf. Process. Syst.* 2012, pp. 2843–2851. ISBN: 9781627480031. DOI: 10.1.1.300.2221.

[30] Richard W. Cole, Tushare Jinadasa, and Claire M. Brown. “Measuring and interpreting point spread functions to determine confocal microscope resolution and ensure quality control”. In: *Nat. Protoc.* 6.12 (2011), pp. 1929–1941. ISSN: 17542189. DOI: 10.1038/nprot.2011.407.

[31] Napari contributors. *napari: a multi-dimensional image viewer for python*. 2019. DOI: doi:10.5281/zenodo.3555620. URL: https://napari.org/.

[32] Corinna Cortes and Vladimir Vapnik. “Support-Vector Networks”. In: *Mach. Learn.* 20 (1995), pp. 273–297. ISSN: 08859000. DOI: 10.1109/64.163674.

[33] Andrea Cossarizza et al. “Guidelines for the use of flow cytometry and cell sorting in immunological studies”. In: *Eur. J. Immunol.* 47.10 (2017), pp. 1584–1797. ISSN: 15214141. DOI: 10.1002/eji.201646632.

[34] William J. Costerton et al. “Bacterial Biofilms in Nature and Disease”. In: *Annu. Rev. Microbiol.* 41.1 (1987), pp. 435–464. ISSN: 0066-4227.

[35] Jean Cousty et al. “Watershed cuts: Minimum spanning forests and the drop of water principle”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 31.8 (2009), pp. 1362–1374. ISSN: 01628828. DOI: 10.1109/TPAMI.2008.173.

[36] John C. Crocker and David G. Grier. “Methods of digital video microscopy for colloidal studies”. In: *J. Colloid Interface Sci.* 179.1 (1996), pp. 298–310. ISSN: 00219797. DOI: 10.1006/jcis.1996.0217.

[37] Holger Daims, Sebastian Lücker, and Michael Wagner. “daime, a novel image analysis program for microbial ecology and biofilm research”. In: *Environ. Microbiol.* 8.2 (2006), pp. 200–213. DOI: 10.1111/j.1462-2920.2005.00880.x.

[38] Francisco Díaz-Pascual et al. “Breakdown of Vibrio cholerae biofilm architecture induced by antibiotics disrupts community barrier function”. In: *Nat. Microbiol.* 4.12 (2019), pp. 2136–2145. ISSN: 20585276. DOI: 10.1038/s41564-019-0579-2.

[39] Anna Dragoš et al. “Division of Labor during Biofilm Matrix Production”. In: *Curr. Biol.* 28.12 (2018), 1903–1913.e5. ISSN: 09609822. DOI: 10.1016/j.cub.2018.04.046.

[40] Knut Drescher. “Bakterielle Multizellularität in Biofilmen”. In: *BioSpektrum* 25.3 (2019), pp. 258–260. ISSN: 18686249. DOI: 10.1007/s12268-019-1038-6.

[41] Knut Drescher et al. "Architectural transitions in Vibrio cholerae biofilms at single-cell resolution". In: *Proc. Natl. Acad. Sci.* 113.14 (2016), E2066–E2072. ISSN: 0027-8424. DOI: 10.1073/pnas.1601702113.

[42] Kenneth W. Dunn et al. "DeepSynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data". In: *Sci. Rep.* 9.1 (2019), pp. 1–15. ISSN: 20452322. DOI: 10.1038/s41598-019-54244-5.

[43] Stephen P. Dzul et al. "Contribution of the Klebsiella pneumoniae capsule to bacterial aggregate and biofilm microstructures". In: *Appl. Environ. Microbiol.* 77.5 (2011), pp. 1777–1782. ISSN: 00992240. DOI: 10.1128/AEM.01752-10.

[44] Harry Eagle. "Amino Acid Metabolism in Mammalian Cell Cultures". In: *Science (80-.).* 130.3373 (1956), pp. 432–437. DOI: 10.1126/science.130.3373.432.

[45] Arthur D. Edelstein et al. "Advanced methods of microscope control using μManager software". In: *J. Biol. Methods* 1.2 (2014), p. 10. ISSN: 2326-9901. DOI: 10.14440/jbm.2014.36.

[46] Dennis Eschweiler et al. "Spherical Harmonics for Shape-Constrained 3D Cell Segmentation". In: *arXiv* 3 (2020), pp. 1–5. arXiv: 2010.12369.

[47] Renaud Escudié et al. "Control of start-up and operation of anaerobic biofilm reactors: An overview of 15 years of research". In: *Water Res.* 45.1 (2011), pp. 1–10. ISSN: 00431354. DOI: 10.1016/j.watres.2010.07.081.

[48] Thorsten Falk et al. "U-Net: deep learning for cell counting, detection, and morphometry". In: *Nat. Methods* 16.1 (2019), pp. 67–70. ISSN: 15487105. DOI: 10.1038/s41592-018-0261-2.

[49] L. Fei-Fei, J. Deng, and K. Li. "ImageNet: Constructing a large-scale image database". In: *J. Vis.* 9.8 (2010), pp. 1037–1037. DOI: 10.1167/9.8.1037.

[50] Hans Curt Flemming and Stefan Wuertz. "Bacteria and archaea on Earth and their abundance in biofilms". In: *Nat. Rev. Microbiol.* 17.4 (2019), pp. 247–260. ISSN: 17401534. DOI: 10.1038/s41579-019-0158-9.

[51] Jiunn CN. Fong et al. "Structural dynamics of RbmA governs plasticity of Vibrio cholerae biofilms". In: *Elife* 6 (2017), pp. 1–22. ISSN: 2050084X. DOI: 10.7554/eLife.26163.

[52] W. C. Fuqua, S. C. Winans, and E. P. Greenberg. "Quorum sensing in bacteria: The LuxR-LuxI family of cell density- responsive transcriptional regulators". In: *J. Bacteriol.* 176.2 (1994), pp. 269–275. ISSN: 00219193. DOI: 10.1128/jb.176.2.269-275.1994.

[53] A. L. Gallego-Hernandez et al. "Upregulation of virulence genes promotes Vibrio cholerae biofilm hyperinfectivity". In: *Proc. Natl. Acad. Sci. U. S. A.* 117.20 (2020), pp. 11010–11017. ISSN: 10916490. DOI: 10.1073/pnas.1916571117.

[54] Ross Girshick. "Fast R-CNN". In: *Proc. IEEE Int. Conf. Comput. Vis.* 2015 Inter (2015), pp. 1440–1448. ISSN: 15505499. DOI: 10.1109/ICCV.2015.169. arXiv: 1504.08083.

[55] Rafael C. Gonzales and Richard E. Woods. *Digital image processing*. 3rd ed. Prentice hall New Jersey, 2002.

[56] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[57] Paul C. Goodwin. *Quantitative deconvolution microscopy*. 1st ed. Vol. 123. Elsevier Inc., 2014, pp. 177–192. ISBN: 9780124201385. DOI: 10.1016/B978-0-12-420138-5.00010-0.

[58] Priya Goyal et al. "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour". In: *arXiv* (2017). ISSN: 2167-3888. DOI: 10.1561/2400000003. arXiv: 1706.02677.

[59] Elisa T. Granato and Kevin R. Foster. "The Evolution of Mass Cell Suicide in Bacterial Warfare". In: *Curr. Biol.* 30.14 (2020), 2836–2843.e3. ISSN: 18790445. DOI: 10.1016/j.cub.2020.05.007.

[60] Manuel Guizar-Sicairos, Samuel T Thurman, and James R Fienup. "Efficient subpixel image registration algorithms". In: *Opt. Lett.* 33.2 (2008), pp. 156–158. ISSN: 0146-9592. DOI: 10.1364/OL.33.000156. arXiv: 78 [0146-9592].

[61] Min Guo et al. "Rapid image deconvolution and multiview fusion for optical microscopy". In: *Nat. Biotechnol.* 38.November (2020). ISSN: 15461696. DOI: 10.1038/s41587-020-0560-x.

[62] Matthias G. Haberl et al. "CDeep3M—Plug-and-Play cloud-based deep learning for image segmentation". In: *Nat. Methods* 15.9 (2018), pp. 677–680. ISSN: 15487105. DOI: 10.1038/s41592-018-0106-z.

[63] Alon Halevy, Peter Norvig, and Fernando Pereira. "The Unreasonable Effectiveness of data". In: *IEEE Intell. Syst.* 24.2 (2009), pp. 8–12. DOI: 10.1109/MIS.2009.36.

[64] Raimo Hartmann. "Examining Uptake of Nanomaterials by Eukaryotic Cells with Digital Image Cytometry". Dissertation. Philipps-Universität Marburg, 2015, pp. 1–213. DOI: 10.17192/z2015.0375.

[65] Raimo Hartmann et al. "BiofilmQ: quantitative image analysis of microbial biofilm communities". In: *bioRxiv Prepr.* (2019). DOI: 10.1101/735423.

[66] Raimo Hartmann et al. "Emergence of three-dimensional order and structure in growing biofilms". In: *Nat. Phys.* 15.3 (2019), pp. 251–256. ISSN: 17452481. DOI: 10.1038/s41567-018-0356-9.

[67] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conf. Comput. Vis. Pattern Recognit.* 2016, pp. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

[68] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proc. IEEE Int. Conf. Comput. Vis.* 2015 Inter (2015), pp. 1026–1034. ISSN: 15505499. DOI: 10.1109/ICCV.2015.123. arXiv: 1502.01852.

[69] Kaiming He et al. “Mask R-CNN”. In: *Proc. IEEE Int. Conf. Comput. Vis.* 2017, pp. 2980–2988. ISBN: 9781538610329. DOI: 10.1109/ICCV.2017.322. arXiv: 1703.06870.

[70] Rainer Heintzmann and Christoph G. Cremer. “Laterally modulated excitation microscopy: improvement of resolution by using a diffraction grating”. In: *Proc. SPIE - Int. Soc. Opt. Eng.* Vol. 3568. 1999, p. 185. DOI: 10.1117/12.336833.

[71] Stefan W. Hell and Jan Wichmann. “Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy”. In: *Opt. Lett.* 19.11 (1994), p. 780. ISSN: 0146-9592. DOI: 10.1364/ol.19.000780.

[72] A. Heydorn et al. “Experimental reproducibility in flow-chamber biofilms”. In: *Microbiology* 146.10 (2000), pp. 2409–2415. ISSN: 13500872. DOI: 10.1099/00221287-146-10-2409.

[73] Arne Heydorn et al. “Quantication of biofilm structures by the novel computer program”. In: *Microbiology* 146.10 (2000), pp. 2395–2407. ISSN: 1350-0872. DOI: 10.1099/00221287-146-10-2395.

[74] Geoffrey E. Hinton and Ruslan Salakhutdinov. “Reducing the Dimensionality of Data with Neural Networks”. In: *Science (80-.).* 313.July (2006), pp. 504–507. DOI: 10.1126/science.1127647.

[75] David Joon Ho et al. “Nuclei Segmentation of Fluorescence Microscopy Images Using Three Dimensional Convolutional Neural Networks”. In: *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.* 2017-July (2017), pp. 834–842. ISSN: 21607516. DOI: 10.1109/CVPRW.2017.116.

[76] Tin Kam Ho. “Random Decision Forests”. In: *Proc. 3rd Int. Conf. Doc. Anal. Recognit.* 1995, pp. 278–282. ISBN: 0818671289. DOI: 10.1109/ICDAR.1995.598994.

[77] Reka Hollandi et al. “nucleAIzer: A Parameter-free Deep Learning Framework for Nucleus Segmentation Using Image Style Transfer”. In: *Cell Syst.* 10.5 (2020), 453–458.e6. ISSN: 24054720. DOI: 10.1101/580605. URL: http://dx.doi.org/10.1016/j.cels.2020.04.003.

[78] Timothy J. Holmes. “Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach: errata”. In: *J. Opt. Soc. Am. A* 9.11 (1992), p. 2097. ISSN: 1084-7529. DOI: 10.1364/josaa.9.002097.

[79] Jan Huisken et al. “Optical sectioning deep inside live embryos by selective plane illumination microscopy”. In: *Science (80-.).* 305.5686 (2004), pp. 1007–1009. ISSN: 00368075. DOI: 10.1126/science.1100035.

[80] Paul Jaccard. “the Distribution of the Flora in the Alpine Zone.” In: *New Phytol.* 11.2 (1912), pp. 37–50. ISSN: 14698137. DOI: 10.1111/j.1469-8137.1912.tb05611.x.

[81] Michał Januszewski et al. “High-precision automated reconstruction of neurons with flood-filling networks”. In: *Nat. Methods* 15.8 (2018), pp. 605–610. ISSN: 15487105. DOI: 10.1038/s41592-018-0049-4.

[82] Hannah Jeckel and Knut Drescher. "Advances and opportunities in image analysis of bacterial cells and communities". In: *FEMS Microbiol. Rev.* 11 (2020). DOI: 10.1093/femsre/fuaa062.

[83] Hannah Jeckel et al. "Learning the space-time phase diagram of bacterial swarm expansion". In: *Proc. Natl. Acad. Sci.* 116.5 (2019), pp. 1489–1494. ISSN: 0027-8424. DOI: 10.1073/pnas.1811722116.

[84] Eric Jelli et al. *BiofilmQ Online Documentation*. 2020. URL: https://drescherlab.org/data/biofilmQ/docs.

[85] James Jonkman et al. "Tutorial: guidance for quantitative confocal microscopy". In: *Nat. Protoc.* 15.5 (2020), pp. 1585–1611. ISSN: 17502799. DOI: 10.1038/s41596-020-0313-9.

[86] Taehun Kim et al. "Active learning for accuracy enhancement of semantic segmentation with CNN-corrected label curations: Evaluation on kidney segmentation in abdominal CT". In: *Sci. Rep.* 10.1 (2020), pp. 1–7. ISSN: 20452322. DOI: 10.1038/s41598-019-57242-9.

[87] Alexander Kirillov et al. "Panoptic segmentation". In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2019-June (2019), pp. 9396–9405. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00963. arXiv: 1801.00868.

[88] Andreas Klöckner et al. "PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation". In: *Parallel Comput.* 38.3 (2012), pp. 157–174. ISSN: 01678191. DOI: 10.1016/j.parco.2011.09.001. arXiv: 0911.3456.

[89] Philip Kollmannsberger et al. "The small world of osteocytes: Connectomics of the lacuno-canalicular network in bone". In: *New J. Phys.* 19.7 (2017). ISSN: 13672630. DOI: 10.1088/1367-2630/aa764b.

[90] Ksenia Konyushkova, Sznitman Raphael, and Pascal Fua. "Learning active learning from data". In: *Adv. Neural Inf. Process. Syst.* Vol. 30. Curran Associates, Inc., 2017, pp. 4226–4236. arXiv: 1703.03365.

[91] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. "Geometry in active learning for binary and multi-class image segmentation". In: *Comput. Vis. Image Underst.* 182 (2019), pp. 1–16. ISSN: 1090235X. DOI: 10.1016/j.cviu.2019.01.007. arXiv: 1606.09029.

[92] Caitlin H. Kowalski et al. "Fungal biofilm architecture produces hypoxic microenvironments that drive antifungal resistance". In: *Proc. Natl. Acad. Sci. U. S. A.* 117.36 (2020), pp. 22473–22483. ISSN: 10916490. DOI: 10.1073/pnas.2003700117.

[93] Alex Krizhevsky. "One weird trick for parallelizing convolutional neural networks". In: *arXiv* (2014). arXiv: 1404.5997.

[94] Alex Krizhevsky and Geoffrey E. Hinton. "Learning Multiple Layers of Features from Tiny Images". In: (2009). ISSN: 00012475.

[95] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Commun. ACM* 60.6 (2017), pp. 84–90. DOI: 10.1201/9781420010749.

[96] H W Kuhn. "The Hungarian method for the assignment problem". In: *Nav. Res. Logist. Q.* 2.1-2 (1955), pp. 83–97. DOI: 10.1002/nav.3800020109.

[97] Abhishek Kumar et al. "Dual-view plane illumination microscopy for rapid and spatially isotropic imaging". In: *Nat. Protoc.* 9.11 (2014), pp. 2555–2573. ISSN: 17502799. DOI: 10.1038/nprot.2014.172.

[98] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. ISSN: 14764687. DOI: 10.1038/nature14539.

[99] Yann LeCun et al. "Handwritten Digit Recognition with a Back-Propagation Network". In: *Adv. Neural Inf. Process. Syst.* (1990), pp. 396–404. ISSN: 1524-4725. DOI: 10.1111/dsu.12130. arXiv: 1004.3732.

[100] Calvin K. Lee et al. "Multigenerational memory and adaptive adhesion in early bacterial biofilm communities". In: *Proc. Natl. Acad. Sci.* (2018), p. 201720071. ISSN: 0027-8424. DOI: 10.1073/pnas.1720071115.

[101] Ta-Chih Lee and Rangasami L Kashyap. "Building Skeleton Models via 3-D Medial Surface/ Axis Thinning Algorithms". In: *Graph. Model. Image Process.* 56.6 (1994), pp. 462–478.

[102] Keith A. Lidke et al. "Superresolution by localization of quantum dots using blinking statistics". In: *Opt. Express* 13.18 (2005), p. 7052. ISSN: 1094-4087. DOI: 10.1364/opex.13.007052.

[103] Tsung-yi Lin et al. "Microsoft COCO : Common Objects in Context". In: *Eur. Conf. Comput. Vis.* Springer, 2014, pp. 740–755. arXiv: arXiv:1405.0312v3.

[104] Melissa Linkert et al. "Metadata matters: Access to image data in the real world". In: *J. Cell Biol.* 189.5 (2010), pp. 777–782. ISSN: 00219525. DOI: 10.1083/jcb.201004104.

[105] Jintao Liu et al. "Metabolic co-dependence gives rise to collective oscillations within biofilms". In: *Nature* 523.7562 (2015), pp. 550–554. ISSN: 14764687. DOI: 10.1038/nature14660.

[106] Francesco Locatello et al. "Object-Centric Learning with Slot Attention". In: *arXiv* NeurIPS (2020), pp. 1–27. arXiv: 2006.15055.

[107] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: (2014). ISSN: 01628828. DOI: 10.1109/TPAMI.2016.2572683. arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[108] F. R. S. Lord Rayleigh. "Investigations in Optics, with special reference to the Spectroscope". In: *Philos. Mag. J. Sci.* 8.49 (1879), pp. 261–274. DOI: 10.1080/14786447908639684.

[109] D. Marr and E. Hildreth. "Theory of edge detection". In: *Proc. R. Soc. London - Biol. Sci.* 207.1167 (1980), pp. 187–217. ISSN: 09628452. DOI: 10.1098/rspb.1980.0020.

[110] Thomas J. Marrie, Joyce Nelligan, and William J. Costerton. "A scanning and transmission electron microscopic study of an infected endocardial pacemaker lead". In: *Circulation* 66.6 (1982), pp. 1339–1341. ISSN: 0009-7322. DOI: 10.1161/01.CIR.66.6.1339.

[111] Ricardo Martínez-garcía et al. "Cell adhesion and fluid flow jointly initiate genotype spatial distribution in biofilms Short title : Cell adhesion and fluid flow in biofilm early development". In: *PLOS Comput. Biol.* 14.4 (2018), pp. 1–32. ISSN: 1553-7358. DOI: 10.1101/243055.

[112] Dominic Masters and Carlo Luschi. "Revisiting Small Batch Training for Deep Neural Networks". In: *arXiv* (2018), pp. 1–18. arXiv: 1804.07612.

[113] Katie McDole et al. "In Toto Imaging and Reconstruction of Post-Implantation Mouse Development at the Single-Cell Level". In: *Cell* 175.3 (2018), 859–876.e33. ISSN: 00928674. DOI: 10.1016/j.cell.2018.09.031.

[114] Scott Mayer McKinney et al. "International evaluation of an AI system for breast cancer screening". In: *Nature* 577.7788 (2020), pp. 89–94. ISSN: 14764687. DOI: 10.1038/s41586-019-1799-6.

[115] Karin L Meibom et al. "The Vibrio cholerae chitin utilization program". In: *Proc. Natl. Acad. Sci.* 101.8 (2004), pp. 2524–2529. DOI: 10.1073/pnas.0308707101.

[116] Erik Meijering. "Cell Segmentation: 50 Years Down the Road". In: *IEEE Signal Process. Mag.* 29.5 (2012), pp. 140–145. DOI: 10.1109/MSP.2012.2204190.

[117] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533. ISSN: 0028-0836. DOI: 10.1038/nature14236. arXiv: 1312.5602.

[118] Erick Moen et al. "Deep learning for cellular image analysis". In: *Nat. Methods* 16.12 (2019), pp. 1233–1246. ISSN: 15487105. DOI: 10.1038/s41592-019-0403-1.

[119] All Mohraz and Michael J. Solomon. "Direct visualization of colloidal rod assembly by confocal microscopy". In: *Langmuir* 21.12 (2005), pp. 5298–5306. ISSN: 07437463. DOI: 10.1021/la046908a.

[120] Carey D. Nadell, Knut Drescher, and Kevin R. Foster. "Spatial structure, cooperation and competition in biofilms". In: *Nat. Rev. Microbiol.* 14.9 (2016), pp. 589–600. ISSN: 17401534. DOI: 10.1038/nrmicro.2016.84.

[121] K. H. Nealson, T. Platt, and J. W. Hastings. "Cellular control of the synthesis and activity of the bacterial luminescent system." In: *J. Bacteriol.* 104.1 (1970), pp. 313–322. ISSN: 00219193. DOI: 10.1128/jb.104.1.313-322.1970.

[122] Jakub Nedbal. *writeFCS(fname, DATA, TEXT, OTHER)*. 2015. URL: https://de.mathworks.com/matlabcentral/fileexchange/42603-writefcs-fname-data-text-other.

[123] Niklas Netter. "Anwendung von Deep Learning zur Bildverarbeitung von Biofilmen". Bachelor Thesis. Philipps-Universität Marburg, 2018, pp. 1–81.

[124] Thomas R. Neu et al. "Advanced imaging techniques for assessment of structure, composition and function in biofilm systems". In: *FEMS Microbiol. Ecol.* 72.1 (2010), pp. 1–21. ISSN: 01686496. DOI: 10.1111/j.1574-6941.2010.00837.x.

[125] Dao Nguyen et al. "Active Starvation Responses Mediate Antibiotic Tolerance in Biofilms and Nutrient-Limited Bacteria". In: *Science (80-.).* 334.6058 (2011), pp. 982–986. DOI: 10.1126/science.1211037.

[126] Nuno M. Oliveira, Kevin R. Foster, and William M. Durham. "Single-cell twitching chemotaxis in developing biofilms". In: *Proc. Natl. Acad. Sci.* 113.23 (2016), pp. 6532–6537. ISSN: 0027-8424. DOI: 10.1073/pnas.1600760113.

[127] Masafumi Oshiro. "Cooled CCD versus intensified cameras for low-light video - Applications and relative advantages". In: *Methods Cell Biol.* 56 (1998), pp. 45–62. ISSN: 0091679X. DOI: 10.1016/s0091-679x(08)60420-3.

[128] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Trans. Syst. Man. Cybern.* SMC-9.1 (1979), pp. 62–66. ISSN: 0018-9472. DOI: 10.1109/TSMC.1979.4310076.

[129] Simon B. Otto et al. "Privatization of Biofilm Matrix in Structurally Heterogeneous Biofilms". In: *mSystems* 5.4 (2020), pp. 1–12. ISSN: 23795077. DOI: 10.1128/msystems.00425-20.

[130] Krishnan Padmanabhan, William F. Eddy, and Justin C. Crowley. "A novel algorithm for optimal image thresholding of biological data". In: *J. Neurosci. Methods* 193.2 (2010), pp. 380–384. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2010.08.031.

[131] Pitch Patarasuk and Xin Yuan. "Bandwidth optimal all-reduce algorithms for clusters of workstations". In: *J. Parallel Distrib. Comput.* 69.2 (2009), pp. 117–124. ISSN: 07437315. DOI: 10.1016/j.jpdc.2008.09.002.

[132] James B. Pawley. "Fundamental limits in confocal microscopy". In: *Handb. Biol. Confocal Microsc. Third Ed.* (2006), pp. 20–42. DOI: 10.1007/978-0-387-45524-2_2.

[133] Caitlin R. Proctor et al. "Biofilms in shower hoses". In: *Water Res.* 131.15 (2018), pp. 274–286. ISSN: 18792448. DOI: 10.1016/j.watres.2017.12.027.

[134] Boyang Qin et al. "Cell position fates and collective fountain flow in bacterial biofilms revealed by light-sheet microscopy". In: *Science (80-.).* 8501.June (2020), eabb8501. ISSN: 0036-8075. DOI: 10.1126/science.abb8501.

[135] Rajat Raina, Anand Madhavan, and Andrew Y. Ng. "Large-scale deep unsupervised learning using graphics processors". In: *ACM Int. Conf. Proceeding Ser.* 382 (2009). DOI: 10.1145/1553374.1553486.

[136] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.6 (2017), pp. 1137–1149. ISSN: 01628828. DOI: 10.1109/TPAMI.2016.2577031. arXiv: 1506.01497.

[137] William Hadley Richardson. “Bayesian-Based Iterative Method of Image Restoration”. In: *J. Opt. Soc. Am.* 62.1 (1972), p. 55. ISSN: 0030-3941. DOI: 10.1364/josa.62.000055.

[138] T. W. Ridler and S. Calvard. “Picture Thresholding Using an Iterative Selection Method”. In: *IEEE Transacations Syst. Man, Cybern.* SMC.8 (1978), pp. 630–632. DOI: 10.1109/TSMC.1978.4310039.

[139] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Med. Image Comput. Comput. Interv.* 2015, pp. 234–241. ISBN: 9783319245737. DOI: 10.1007/978-3-319-24574-4_28. arXiv: 1505.04597.

[140] Sajith Kecheril Sadanandan et al. “Automated Training of Deep Convolutional Neural Networks for Cell Segmentation”. In: *Sci. Rep.* 7.1 (2017), pp. 1–7. ISSN: 20452322. DOI: 10.1038/s41598-017-07599-6.

[141] Karin Sauer et al. “Pseudomonas aeruginosa Displays Multiple Phenotypes during Development as a Biofilm”. In: *J. Bacteriol.* 184.4 (2002), pp. 1140–1154. DOI: 10.1128/JB.184.4.1140-1154.2002.

[142] Johannes Schindelin et al. “Fiji: An open-source platform for biological-image analysis”. In: *Nat. Methods* 9.7 (2012), pp. 676–682. ISSN: 15487091. DOI: 10.1038/nmeth.2019.

[143] Uwe Schmidt et al. “Cell detection with star-convex polygons”. In: *Lect. Notes Comput. Sci.* 11071 (2018), pp. 265–273. ISSN: 16113349. DOI: 10.1007/978-3-030-00934-2_30. arXiv: 1806.03535.

[144] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualitation Toolkit.* 4th ed. Kitware, Inc, 2006. ISBN: 978-1-930934-19-1.

[145] Andrew W. Senior et al. “Improved protein structure prediction using potentials from deep learning”. In: *Nature* 577.7792 (2020), pp. 706–710. ISSN: 14764687. DOI: 10.1038/s41586-019-1923-7.

[146] Alexander Sergeev and Mike Del Balso. “Horovod: fast and easy distributed deep learning in TensorFlow”. In: *arXiv* (2018). arXiv: 1802.05799.

[147] Connor Shorten and Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. In: *J. Big Data* 6.1 (2019). ISSN: 21961115. DOI: 10.1186/s40537-019-0197-0.

[148] Abhishek Shrivastava et al. “Cargo transport shapes the spatial organization of a microbial community”. In: *Proc. Natl. Acad. Sci. U. S. A.* 115.34 (2018), pp. 8633–8638. ISSN: 10916490. DOI: 10.1073/pnas.1808966115.

[149] David Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science (80-.).* 362.6419 (2018), pp. 1140–1144. ISSN: 10959203. DOI: 10.1126/science.aar6404.

[150] Emilia L Simmons et al. "Biofilm Structure Promotes Coexistence of Phage-Resistance and Phage-Susceptible Bacteria". In: *mSystems* 5.3 (2020), pp. 1–17. DOI: 10.1128/mSystems.00877-19.

[151] Praveen K. Singh et al. "Vibrio cholerae Combines Individual and Collective Sensing to Trigger Biofilm Dispersal". In: *Curr. Biol.* 27.21 (2017), 3359–3366.e7. ISSN: 09609822. DOI: 10.1016/j.cub.2017.09.041.

[152] Christoph Sommer et al. "Ilastik: Interactive learning and segmentation toolkit". In: *Eighth IEEE Int. Symp. Biomed. Imaging* 1 (2011), pp. 230–233. ISSN: 1424441277. DOI: 10.1109/ISBI.2011.5872394.

[153] Josef Spidlen et al. "Data File Standard for Flow Cytometry, Version FSC 3.1". In: *Cytom. A* 77.1 (2010), pp. 97–100. DOI: 10.1002/cyto.a.20825.

[154] Elizabeth J. Stewart et al. "Role of environmental and antibiotic stress on Staphylococcus epidermidis biofilm microstructure". In: *Langmuir* 29.23 (2013), pp. 7017–7024. ISSN: 07437463. DOI: 10.1021/la401322k.

[155] Philip S. Stewart and Michael J. Franklin. "Physiological heterogeneity in biofilms". In: *Nat. Rev. Microbiol.* 6.3 (2008), pp. 199–210. ISSN: 17401526. DOI: 10.1038/nrmicro1838. arXiv: NIHMS150003.

[156] Carsen Stringer et al. "Cellpose: a generalist algorithm for cellular segmentation". In: *bioRxiv* (2020), pp. 1–19. DOI: 10.1101/2020.02.02.931238.

[157] Hao Su, Jia Deng, and Li Fei-Fei. "Crowdsourcing annotations for visual object detection". In: *AAAI Work. - Tech. Rep.* WS-12-08 (2012), pp. 40–46. DOI: 10.1.1.309.4396.

[158] Chen Sun et al. "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era". In: *Proc. IEEE Int. Conf. Comput. Vis.* October (2017), pp. 843–852. ISSN: 15505499. DOI: 10.1109/ICCV.2017.97. arXiv: 1707.02968.

[159] Ludvík Tesař et al. "Medical image analysis of 3D CT images based on extension of Haralick texture features". In: *Comput. Med. Imaging Graph.* 32.6 (2008), pp. 513–520. ISSN: 08956111. DOI: 10.1016/j.compmedimag.2008.05.005.

[160] Vladimír Ulman et al. "An objective comparison of cell-tracking algorithms". In: *Nat. Methods* 14 (2017), pp. 1141–1152. DOI: 10.1038/nmeth.4473.

[161] Guido Van Rossum and Fred L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995. URL: https://www.python.org/.

[162] David A. Van Valen et al. "Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments". In: *PLoS Comput. Biol.* 12.11 (2016), pp. 1–24. ISSN: 15537358. DOI: 10.1371/journal.pcbi.1005177.

[163] Tomas Vicar et al. "Cell segmentation methods for label-free contrast microscopy: review and comprehensive comparison". In: *BMC Bioinformatics* 20.1 (2019), p. 360. ISSN: 1471-2105. DOI: 10.1186/s12859-019-2880-8.

[164] Lucia Vidakovic et al. "Dynamic biofilm architecture confers individual and collective mechanisms of viral protection". In: *Nat. Microbiol.* 3.1 (2017), pp. 26–31. ISSN: 20585276. DOI: 10.1038/s41564-017-0050-1.

[165] Martin Vorregaard. "Comstat2 - a modern 3D image analysis environment for biofilms". Master Thesis. Technical University of Denmark, 2008, pp. 1–72.

[166] W. Wallace, L. H. Schaefer, and J. R. Swedlow. "A workingperson's guide to deconvolution in light microscopy". In: *Biotechniques* 31.5 (2001), pp. 1076–1097. ISSN: 07366205. DOI: 10.2144/01315bi01.

[167] J. Wang et al. "Bact-3D: A level set segmentation approach for dense multi-layered 3D bacterial biofilms". In: *Proc. - Int. Conf. Image Process. ICIP* September (2018), pp. 330–334. ISSN: 15224880. DOI: 10.1109/ICIP.2017.8296297.

[168] J. Wang et al. "Lcuts: Linear Clustering of Bacteria Using Recursive Graph Cuts". In: *2019 IEEE Int. Conf. Image Process.* 2019, pp. 1575–1579. DOI: 10.1109/icip.2019.8803064. arXiv: arXiv:1902.00166v3.

[169] Wenzhe Wang et al. "Nodule-Plus R-CNN and Deep Self-Paced Active Learning for 3D Instance Segmentation of Pulmonary Nodules". In: *IEEE Access* 7 (2019), pp. 128796–128805. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2939850.

[170] D. Webb et al. "Assessing technician effects when extracting quantities from microscope images". In: *J. Microbiol. Methods* 53.1 (2003), pp. 97–106. ISSN: 01677012. DOI: 10.1016/S0167-7012(02)00228-2.

[171] Martin Weigert et al. "Content-Aware Image Restoration: Pushing the Limits of Fluorescence Microscopy". In: *Nat. Methods* 15.12 (2018), pp. 1090–1097. ISSN: 15487105. DOI: 10.1101/236463.

[172] Martin Weigert et al. "Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy". In: *2020 IEEE Winter Conf. Appl. Comput. Vis.* March. 2020. ISBN: 9781728165530. DOI: 10.1109/WACV45572.2020.9093435. arXiv: 1908.03636.

[173] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press, 1950.

[174] Stephan Wimmi et al. "Dynamic relocalization of the cytosolic type III secretion system components prevents premature protein secretion at low external pH". In: (2019). DOI: 10.1101/869214.

[175] Steffen Wolf, Fred A. Hamprecht, and Jan Funke. "Instance Separation Emerges from Inpainting". In: *arXiv* (2020). arXiv: 2003.00891.

[176] Steffen Wolf et al. "Learned Watershed: End-to-End Learning of Seeded Segmentation". In: *Proc. IEEE Int. Conf. Comput. Vis.* 2017-Octob (2017), pp. 2030–2038. ISSN: 15505499. DOI: 10.1109/ICCV.2017.222. arXiv: 1704.02249.

[177] Adrian Wolny et al. "Accurate and versatile 3D segmentation of plant tissues at cellular resolution". In: *Elife* 9 (2020), pp. 1–34. ISSN: 2050084X. DOI: 10.7554/eLife.57613.

[178] Yicong Wu et al. "Spatially isotropic four-dimensional imaging with dual-view plane illumination microscopy". In: *Nat. Biotechnol.* 31.11 (2013), pp. 1032–1038. ISSN: 10870156. DOI: 10.1038/nbt.2713.

[179] Jing Yan et al. "Vibrio cholerae biofilm growth program and architecture revealed by single-cell live imaging". In: *Proc. Natl. Acad. Sci. U. S. A.* 113.36 (2016), e5337–e5343. ISSN: 10916490. DOI: 10.1073/pnas.1611494113.

[180] Lin Yang et al. "Suggestive annotation: A deep active learning framework for biomedical image segmentation". In: *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 10435 LNCS (2017), pp. 399–407. ISSN: 16113349. DOI: 10.1007/978-3-319-66179-7_46. arXiv: 1706.04737.

[181] Linfeng Yang et al. "NuSeT: A deep learning tool for reliably separating and analyzing crowded cells". In: *PLoS Comput. Biol.* 16.9 (2020), pp. 1–20. ISSN: 15537358. DOI: 10.1371/journal.pcbi.1008193.

[182] David Zamorano-Sánchez et al. "Functional Specialization in Vibrio cholerae Diguanylate". In: *MBio* 10.2 (2019), pp. 1–16. DOI: 10.1128/mBio.00670-19.

[183] Mingxing Zhang et al. "Non-Invasive Single-Cell Morphometry in Living Bacterial Biofilms". In: *bioRxiv* (2020), p. 2020.05.28.120279. DOI: 10.1101/2020.05.28.120279.

[184] Hao Zheng et al. "Biomedical image segmentation via representative annotation". In: *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019* 1 (2019), pp. 5901–5908. ISSN: 2159-5399. DOI: 10.1609/aaai.v33i01.33015901.

List of Figures

List of Tables

Publications

- Topological metric detects hidden order in disordered media D. J. Skinner, B. Song, H. Jeckel, E. Jelli, K. Drescher, J. Dunkel Accepted for publication in Physical Review Letters (2020)
  - Data Analysis: Segmentation of swarming data
- BiofilmQ: a software tool for quantitative image analysis of microbial biofilm communities R. Hartmann*, H. Jeckel*, E. Jelli*, P.K. Singh, S. Vaidya, M. Bayer, D. Rode, L. Vidakovic, F. Díaz-Pascual, J.C.N. Fong, A. Dragoš, O. Besharova, J.G. Thöming, N. Netter, S. Häussler, C.D. Nadell, V. Sourjik, A.T. Kovács, F.H. Yildiz, K. Drescher Accepted for publication in Nature Microbiology (2020). (* equal contribution)
  - Experiments: Gathering microscope data for image cytometry, contributing to flow cytometry data collection
  - Software Design: Co-design GUI, Co-implementation of GUI, feature implementations, optimizing & reviewing code, set up of version control and continuous integration, creation of executables, bug fixing
  - Documentation: Design and writing of online documentation, additions to landing page, contributing video tutorials
  - Data Analysis: Segmentation microscope data, analysis flow cytometry data
  - Manuscript: Contributing to manuscript, creation of figures, creation 3D renderings
- Kin discrimination in social yeast is mediated by cell surface receptors of the Flo11 adhesin family S. Brückner, R. Schubert, T. Kraushaar, R. Hartmann, D. Hoffmann, E. Jelli, K. Drescher, D.J. Müller, L.-O. Essen, H.-U. Mösch eLife 9:e55587 (2020)
  - Data Analysis: Implementation of custom segmentation and quantification pipeline for competition experiments
- Learning the space-time phase diagram of bacterial swarm expansion H. Jeckel, E. Jelli, R. Hartmann, P.K. Singh, R. Mok, J.F. Totz, L. Vidakovic, B. Eckhardt, J. Dunkel, K. Drescher PNAS 116, 1489-1494 (2019).
  - Experiments: Optimizing/ feature additions / bug fixes in adaptive microscope control software, collecting confocal microscope data of reporter strains, design of swarm lag time experiment, collecting microscope data for swarm lag times
  - Data Analysis: Quantification of swarm lag times

Acknowledgements

This thesis would not be possible without the great group of people of at the biofilm research group at the Max Planck Institute for Terrestrial Microbiology in Marburg. First, I would like to thank my supervisor Knut Drescher, for the opportunity to prepare my thesis in his lab and the guidance during the research projects. Second, I would like to thank my thesis advisory committee Victor Sourjik and Peter Lenz for their time and advices during the advisory sessions.

I really enjoyed working among those great colleges I had over the past years: Hannah, Raimo, Sanika, Takuya, Konstantin, Niklas, Daniel, Miriam, Praveen, Eva, Francisco, Mads, Lucia, Keerthana, Sven, Kerstin, Viola, Pauline, and Martin. Only with your diverse skill sets, this was all possible and I never forget you. Thank you, Leonie, Hannah, Sanika, Niklas, and Konstantin for proofreading (parts of) this thesis. The number of language flaws you found in the final stages probably will exceed any number of software bugs I am able to fix in my lifetime.

Finally, yet importantly, I am very thankful to you Leonie, for the support, the advices, the patience and joy you brought into my life.

Appendix

Table S1: Overview of modules which extract global and per-object parameter from segmented objects (Part 1).

Default parameter	
Centroid Coordinate	Position the object centre in x, y, and z-coordinate
Shape_Volume	Volume of segmented object
Intensity_Mean	Mean voxel intensity values of segmented object
Cube_VolumeFraction	Ratio of filled volume in a dissection cube (only for pseudo-cells)
Cube_Surface	Number of voxels of facing a non-segmented voxel for segmented object (only for pseudo-cells)
Cube_CenterCoord	Pseudo-cell centre, regardless of biovolume distribution.
Grid_ID	Global ID of pseudo-cell (constant for constant cube size and constant image size)
ID	Object ID
Minimal rotated bounding box	
MinBoundBox_Width	Width of the minimal rotated bounding box (length of the shortest box edge)
MinBoundBox_Height	Height of the minimal rotated bounding box (length of the second longest box edge)
MinBoundBox_Length	Length of the minimal rotated bounding box (longest edge)
MinBoundBox_Cornerpoints	Coordinates of the box corner points
Size and orientation by ellipsoidal fit	
Shape_Width	Length of shortest ellipsoid axes
Shape_Heigth	Length of the second longest ellipsoid axis
Shape_Length	Length of the longest ellipsoid axis
Orientation_Matrix	Vector directions of the ellipsoid axes
Aspect ratios	
Shape_AspectRatio_LengthToWidth	Ratio of length divided by width
Shape_AspectRatio_HeightToWidth	Ratio of height divided by width
Surface properties	
Surface_LocalRoughness	Number of interface voxels between occupied and unoccupied volume in a sphere around centroid
Surface_PerSubstrateArea	Surface above a small substrate area per substrate area
Surface_LocalThickness	Thickness of the biofilm at the xy position in the substrate
Biofilm_MeanThickness	Average of the thickness for the full biofilm
Biofilm_Roughness	Global roughness of the biofilm
Biofilm_OuterSurface	Number of all voxels which are facing non-segmented volumes
Substrate Area	
Architecture_LocalSubstrateArea	Number of voxels interfacing with the substrate plane per segmented object
Biofilm_SubstrateArea	Number of all voxels interfacing with the substrate plane

Table S2: Overview of modules which extract global and per-object parameter from segmented objects (Part 2).

Global biofilm properties	
Biofilm_Width	Range of x values in the segmented biofilm
Biofilm_Height	Range of y values in the segmented biofilm
Biofilm_Length	Range of z values within the [1%, 99%] percentile rank
Biofilm_BaseEccentricity	Eccentricity of the biofilm/ substrate interface
Biofilm_BaseArea	Area of a fitted ellipse with substrate plane
Biofilm_Volume	Volume of the segmented volume in the biofilm
Biofilm_AspectRatio_HeightToLength	Biofilm height divided by biofilm length
Biofilm_AspectRatio_HeightToWidth	Biofilm height divided by biofilm width
Biofilm_AspectRatio_LengthToWidth	Biofilm length divided by biofilm width
Biofilm_SubstrateArea	Area of the biofilm/ substrate interface
Biofilm_VolumePerSubstrateArea	Global biofilm volume divided by base area of the biofilm
Biofilm_OuterSurfacePerSubstrate	Biofilm surface divided by base area
Biofilm_OuterSurfacePerVolume	Biofilm surface divided by biofilm volume
Alignments	
Alignment_Flow	Orientation of the principal axis of a segmented object with the flow direction
Alignment_Zaxis	Orientation of the principal axis of a segmented object with the z-axis
Alignment_Radial	Orientation of the principal axis of a segmented object in relation to the biofilm radial axis
Convexity	
Shape_Convexity	Volume of a segmented object divided by the volume of its convex hull
Distance to center biofilm	
Distance_ToBiofilmCenter	Distance of an object centroid to the biofilm centre
Distance_ToBiofilmCenterAtSubstrate	Distance of an object centroid to the biofilm centre projected onto the xy-plane
Distance to the nearest neighbour	
Distance_ToNearestObject	Distance to the closest object in a user-defined channel
Distance_ToNearestNeighbor	Distance to the nearest object in the object channel
Inter cell spacing	
Distance_InterCellSpacing_Mean	Mean ray tracing distance from object faces to nearby objects
Distance_InterCellSpacing_Min	Minimal ray tracing distance from object faces to nearby objects
Distance_InterCellSpacing_Variance	Variance of the ray tracing distance to nearby objects
Distance to surface	
Distance_ToSurface	Object distance to the nearest point in the biofilm convex hull
Distance to surface (one side excluded)	
Distance_ToSurfaceSideOnly	Object distance to the nearest point in the biofilm convex hull for partly imaged biofilm
Distance to specific object	
Distance_ToObject_id	Object distance to another user-defined object
Nematic order parameter	
Architecture_NematicOrderParameter	Measurement of alignment with object neighbours
Local density	
Architecture_LocalNumberDensity	Number of object in user-defined vicinity of the object
Architecture_LocalDensity	Filling ratio of a sphere through the object centroid
Unit cell size	
Architecture_UnitCellSize	Size of cell in Voronoi diagram

Table S3: Overview of modules which extract global and per-object parameter from segmented objects (Part 3).

Intensity properties	
Intensity_Mean	Mean voxel intensity values of segmented object
Intensity_Mean_noBackground	Integrated voxel intensity values of segmented object with an automatic threshold subtracted
Intensity_Integrated	Integrated voxel intensity values of segmented object
Intensity_Integrated_noBackground	Integrated voxel intensity values of segmented object with an automatic threshold subtracted
Intensity_Ratio_Mean	Ratio between mean intensities in different channels
Intensity_Ratio_Mean_noBackground	Ratio between mean intensities without background in different channels
Intensity_Ratio_Integrated	Ratio between integrated intensities in different channels
Intensity_Ratio_Integrated_noBackground	Ratio between integrated intensities in different channels without background
Intensity_Shells_Integrated	Integrated intensity in a shell around the segmented object
Intensity_Shells_Integrated_noBackground	Integrated intensity in a shell around the segmented object without background
Intensity_Shells_Mean	Mean intensity in a shell around the segmented object
Intensity_Sheslls_Mean_noBackground	Mean intensity in a shell around the segmented object without background
Number of fluorescent foci	
Foci_Number	Counted number of fluorescent foci inside a object
Foci_Idx	List of foci indices per object
Foci_Intensity	List of foci intensities per object
Foci_Quality	List of foci quality per object
Visualize extra-cellular fluorophores	
vtk-file	Single VTK file which visualizes extracellular fluorophores
Correlation properties	
Correlation_Pearson	Pearson's correlation coefficient between two channels
Correlation_Manders	Manders' overlap coefficient between two channels
Correlation_MandersSplit	The Manders' split coefficient of channel in the other
Correlation_AutoCorrelation	Autocorrelation function in 3D for the full image
Correlation_AutoCorrelation_CorrelationLength2D	Average correlation length (i.e. distance with a 50% drop in the autocorrelation) calculated on every plane in the volume
Correlation_AutoCorrelation_CorrelationLength3D	3D correlation length for the full image
Correlation_AutoCorrelation_Zero2D	Position of the first zero-crossing of the averaged 2D correlation function
Correlation_AutoCorrelation_Zero2D_Substrate	Position of the first zero-crossing of the 2D correlation function in the brightest stack plane
Correlation_AutoCorrelation_Zero3D	Position of the first zero-crossing of the 3D correlation function
Correlation_DensityCorrelation	Density correlation for each object
Correlation_DensityCorrelation_Binary	Density correlation on the full binary image which resulted from the segmentation
Correlation_Local3dOverlap	Volume overlap between object and the objects in a user defined channel
Correlation_LocalOverlapFraction	Normalized volume overlap between object and the objects in a user-defined channel
Biofilm_Overlap	Sum of the individual object overlaps
Biofilm_OverlapFraction	Sum of the individual object overlap fractions
Haralick texture features [159]	
Texture_Haralick_Energy	Haralick energy calculation for pseudo-cells
Texture_Haralick_Entropy	Haralick entropy calculation for pseudo-cells
Texture_Haralick_Correlation	Haralick correlation calculation for pseudo-cells
Texture_Haralick_Contrast	Haralick contrast calculation for pseudo-cells
Texture_Haralick_Homogeneity	Haralick homogeneity calculation for pseudo-cells
Texture_Haralick_Variance	Haralick variance calculation for pseudo-cells
Texture_Haralick_SumMean	Haralick mean sum calculation for pseudo-cells
Texture_Haralick_Inertia	Haralick inertia calculation for pseudo-cells
Texture_Haralick_ClusterShade	Haralick cluster shade calculation for pseudo-cells
Texture_Haralick_ClusterTendency	Haralick cluster tendency calculation for pseudo-cells
Texture_Haralick_MaxProbability	Haralick max probability calculation for pseudo-cells
Texture_Haralick_InverseVariance	Haralick inverse variance calculation for pseudo-cells

Table S4: Overview of modules which extract global and per-object parameter from segmented objects (Part 4).

Tag cells	
custom parameter tag	Custom name tag for objects which fullfil a user-defined condition
Custom parameter	
custom parameter name	Combine existing parameter to a user-defined new parameter
Parameter based on user-defined Matlab script	
as defined in custom script	Use *Matlab* script for an arbitrary complex new parameter
Tracking parameter	
Track_GrowthRate	Calculated volume change along tracked cell path
Track_VolumeLossDispersingCells	Calculate volume which is lost due to dispersing cells

Table S5: Folder structure *BiofilmQ* files. Single files are written in *italic*. Folder structure is indicated by indentions.

File / Folder	**Description**
BiofilmQ.m	Main source file with start script and callbacks
BiofilmQ.fig	GUI definitions and callback conntections
parameter names.xlsx	Overview of current parameter names and naming in old version
readme.txt	Instructions for deployment (auto-generated)
requiredMCRProducts.txt	List of needed Matlab Compiler Runtime modules (auto-generated)
LICENSE	Copyright and terms of use statements
batch processing	Collection of pre-shipped batch processing scripts
batchFiles	Standard batch processing files for image processing steps
analysis	Batch files for the HTML overview generation
biovolume_calculations	Example for custom batch files
custom	Specialised batch files for the generation of shell VTK-files, arithmetic parameter modifications, and replacing white spaces in *Huygens* files
parameter files	Temporary folder for batch parameter files
deployment	Scripts of the creation of executables, license „boiler plates“, code house-keeping, and logos
doc	Files for the HTML documentation website
includes	Main source code folder
additional modules visualization	Folder for experimental lineage tracking, tree plotting and 3D visualization in *BiofilmQ*
trajectory visualization	Functions for lineage trees visualizations based on the single-cell tracking (experimental)
additional modules	Folder for not-yet-published *BiofilmQ* features
cell tracking	Start script for single-cell tracking
ellipse representation	Start script for ellipse representation
huygens deconvolution	Start script for *Huygens* deconvolution features
image series curation	Start script for image series curation as preprocessing step
seeded watershed	Start script, callbacks, GUI definitions, and source files for seeded watershed segmentation
simulations	Start script and source files for simulation support
single cell segmentation	Start script and source files for single cell segmentation
thresholding by slice	Start script, callbacks, GUI definitions, and source files for slice-wise threshold segmentation
additionalCallbacks.m	Experimental callbacks
biofilm analysis	Source code files for the *BiofilmQ* visualization tab
custom scripts	Example scripts to modify the plotting data/ plots before/after plotting
includes	Source code files for plotting
plotting	Source code files for the different plot types
deconvolution	Source code files for the automatic generation of *Hugens* batch files
export	Source code files for CSV-, FCS-, vtk-file export, and automatic *ParaView* rendering
file handling	Source code files for file input/ output from/ to disk
functionality	Helping functions for visualization and housekeeping purposes
help	Visualizations of biofilms for object dissection
image processing	Source code files for segmentation
image registration	Source code files for all image registration
layout	Source code filed for GUI modifications and start splash screen
object processing	Source code files parameter calculations in *BiofilmQ*
performance	Compiled *MatLab* function for performance bottlenecks
tools	Folder for tools which are displayed in a popup window
chooseBiofilm	Semi-automatic biofilm selection dialogue
folderNavigator	Folder listing for batch processing
mergeCells	Tool for merging cube-dissected objects
zSlicer	Sliced 3D representation of a biofilm
tests	Folder for unit tests (experimental)

Table S6: Parameters for the *Huygens* deconvolution for manual data annotation pipeline in Sec. 5.5

Property	**Value**
Back-projected pinhole	250 nm
Pinhole spacing	5 µm
Algorithm	Classic MLE
PSF mode	Theoretical PSF
Max iteration	60
Iteration mode	Optimized
Quality change threshold	0.001 %
SNR	20 20 20 20 2
Background mode	Auto
Background estimation radius	0.7
Relative background	0.0 0.0 0.0 0
Bleaching correction	If possible
Brick mode	Auto
PSFs per brick mode	Auto
PSFs per brick manual mode	1

Curriculum Vitae

Dieser Abschnitt enthält persönliche Daten und ist deshalb nicht Bestandteil der Online-Veröffentlichung.